Gestion de projet informatique

Stéphane Grare





Gestion de projet informatique

Stéphane Grare



PREFACE

On parle tous de projets dans la vie de tous les jours : nos projets de vacances, projets de carrière, projets d'avoir des enfants... Le terme projet est donc un terme du vocabulaire courant, auquel on associe une signification relativement claire et précise : c'est un ensemble d'actions que nous souhaitons entreprendre, pour atteindre un but (devenir parent, embrasser une nouvelle carrière...). En ce sens, le projet est bien « le brouillon de l'avenir » : une ébauche, mais pas encore une réalisation.

Cette notion de projet nous vient du latin « projectum de projicere », qui signifie littéralement « jeter quelque chose vers l'avant ».

Au premier abord, un projet est une chose ou un ensemble de choses que l'on se propose de faire, une intention, une ébauche.

Latins et Anglo-saxons accordent un sens assez différent à la notion de projet. Si pour nous le projet n'est qu'une action ou un ensemble d'actions que l'on projette de réaliser, dans la culture anglo-saxonne le projet désigne une notion concrète, incluant la planification, l'anticipation des risques, les acteurs impliqués... Bref, cette notion recouvre un concept plus précis, concret et pragmatique, qui appelle l'action. Nous parlerons, par la suite, de projet en ce sens.

On dénote, de manière assez intuitive, une notion forte de temporalité dans la notion de projet : un projet est généralement une aventure temporaire (ayant à ce titre un début et une fin). Il ne s'agit donc pas d'un processus répétitif : un projet est unique.

Outre les projets personnels, la majorité des projets impliquent plusieurs personnes (une compagne ou un compagnon pour devenir parent, éventuellement une famille pour partir en vacances...). On parle alors d'acteurs du projet. Ces acteurs constituent autant de ressources du projet.

En plus de ces ressources « humaines », un projet peut nécessiter, dans sa réalisation, des ressources matérielles : une voiture pour partir en vacances, une robe de mariée, des bouteilles de champagne...

L'ensemble de ces ressources représente un coût : Salaires et rémunérations pour les ressources humaines, prix d'achat ou de location pour les ressources matérielles. Un projet fait donc généralement l'objet d'une budgétisation.

Enfin, le projet aboutit, normalement, à la production de résultats matériels et immatériels. On appelle ces résultats des livrables, qui représentent les résultats attendus du projet.

Un projet est une chose ou un ensemble de choses que l'on se propose de faire en un temps donné, mettant en œuvre des ressources humaines et matérielles faisant l'objet d'une budgétisation, et aboutissant à un ensemble de livrables.

TABLE DES MATIÈRES

LE CYCLE DE VIE D'UN PROJET

Un projet est une opération dans laquelle des ressources humaines, financières et matérielles sont organisées d'une façon originale, pour réaliser un ensemble de fournitures, selon des spécifications définies, avec des contraintes de coûts et délais, de façon à obtenir un changement bénéfique défini par des objectifs quantitatifs et qualitatifs.

Un projet est un ensemble unique d'actions coordonnées, avec des dates définies de début et de fin, entreprises par un individu ou une entité pour atteindre des objectifs spécifiés en respectant des paramètres de coûts, délais et performances.

Définition de l'AFNOR : "Un projet est un ensemble d'actions à réaliser pour satisfaire un objectif défini, dans le cadre d'une mission précise, et pour la réalisation desquelles on a identifié non seulement un début, mais aussi une fin".

Définition de l'AFNOR : "La gestion de projet est l'ensemble des méthodes, outils d'évaluation, de planification et d'organisation permettant d'atteindre ses objectifs en respectant les contraintes de performance, de délais, et de coûts."

Le management de projet consiste à planifier, organiser, piloter et maîtriser tous les aspects d'un projet, ainsi que la motivation de tous ceux qui sont impliqués dans le projet et la maîtrise de la relation client, de façon à atteindre les objectifs de façon sûre et dans les critères définis de coûts, délais et performances. Cela inclut les tâches de direction nécessaires aux performances du projet.

Un objectif est une contrainte qui va être « imposée » au système projet afin qu'il se réalise dans un cadre. Ce cadre est imposé par le commanditaire.

Un projet comporte 3 niveaux d'objectifs :

-> Les objectifs de qualité.
-> Les objectifs de temps.
-> Les objectifs de coût.

Objectifs de qualité : Ce sont tous les éléments qui vont « qualifier » le produit qui va sortir du projet. Ces éléments vont constituer les performances du produit ; ce sont ces performances qui vont satisfaire le besoin.

Objectifs de temps : C'est le calendrier dans lequel le projet doit se réaliser. Ce calendrier comporte une date de début du projet, une date de fin du projet, des échéances intermédiaires éventuelles.

Objectifs de coût : C'est la somme du coût des ressources nécessaires à la mise en œuvre du projet.

Un objectif doit répondre à un certain nombre de critères :

-> Il doit être mesurable : car il faut pouvoir le visualiser et le comprendre, il doit donc être quantifié. Cela permettra par ailleurs de savoir si l'objectif a été atteint par la mesure des résultats à la fin du projet.

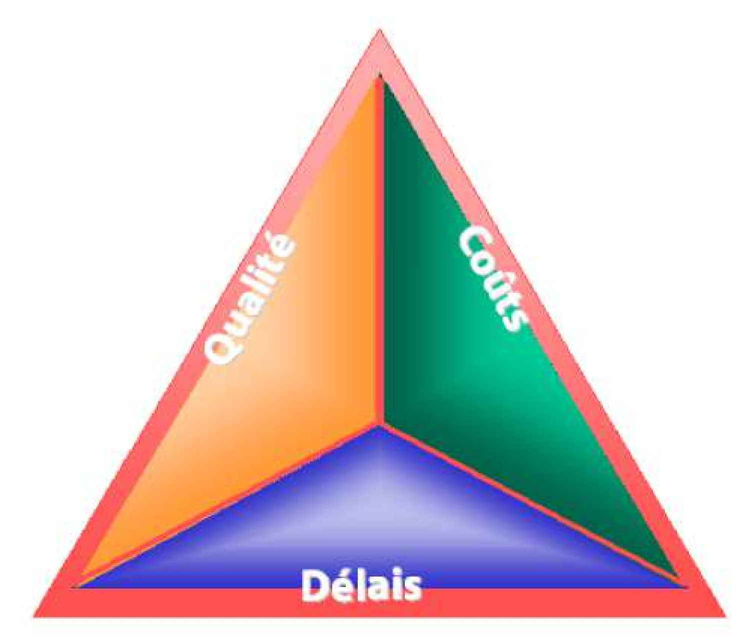
-> Il doit être réalisable : car il faut pouvoir atteindre l'objectif. L'objectif impliquera un engagement du chef de projet ; et pour que celui-ci s'engage sur les objectifs, il faut bien sûr que l'objectif soit atteignable.

-> Il doit pouvoir être négocié : afin d'obtenir un engagement mutuel entre celui qui fixe l'objectif et celui qui se propose de l'atteindre, la négociation s'engage afin d'avoir un accord mutuel qui donne toutes les chances au projet d'aboutir.

-> Il doit être partagé : si l'objectif va être réalisé par un groupe de personnes il faut que cet objectif soit compris par toutes afin qu'il n'y ait aucune ambiguïté entre elles et sur le but à atteindre.

-> Il doit être individualisé : on ne fixe pas un objectif directement à un groupe de personnes. On s'assure que l'objectif a été réparti entre ces personnes et que chacune connaisse sa part d'objectif à atteindre.

Les 3 objectifs Coût/Qualité/Temps sont interdépendants entre eux et interagissent pendant le projet. Si on modifie un seul des objectifs en cours de projet, les deux autres objectifs vont être modifiés.



L'assemblage des 3 objectifs s'appelle le triangle de la performance.

Le cycle de vie d'un projet informatique est composé des étapes et des enchaînements nécessaires pour réaliser le produit ou le service qui font l'objet du projet. Le cycle de vie doit être adapté, en fonction de la complexité du produit à fabriquer. Généralement, il a pour but de participer à la maîtrise les risques de fabrication, ainsi que d'assurer la qualité du produit fini.

MODELES DE DÉVELOPPEMENT

Il existe des "modèles de développement" pour le découpage des projets SI. Un modèle est une abstraction de quelque chose de réel qui permet de comprendre avant de construire ou

de retrouver les informations nécessaires pour effectuer des modifications et extensions. Le modèle simplifie la gestion de la complexité en offrant des points de vue et niveaux d'abstraction plus ou moins détaillés selon les besoins. Il facilite la communication entre les différents intervenants sur le projet. Il supporte la conduite et la gestion des processus de développement et de maintenance.



Les modèles de cycle de vie du logiciel décrivent à un niveau très abstrait et idéalisé les différentes manières d'organiser la production. Les étapes, leur ordonnancement, et parfois les critères pour passer d'une étape à une autre sont explicitées (critères de terminaison d'une étape - revue de documents -, critères de choix de l'étape suivante, critères de démarrage d'une étape).

Le cycle de vie d'un projet informatique est composé des étapes et des enchaînements nécessaires pour réaliser le produit ou le service qui font l'objet du projet. Le cycle de vie doit être adapté, en fonction de la complexité du produit à fabriquer. Généralement, il a pour but de participer à la maîtrise les risques de fabrication, ainsi que d'assurer la qualité du produit fini.

On appelle "cycle de vie canonique" le cycle de vie de base d'un projet informatique. En fonction de l'enchaînement des phases de projet, on débouche sur un cycle de vie en cascade, en V, ou évolutif. D'autre part, en fonction de la complexité du résultat à produire, s'il s'agit d'un projet ou d'un programme, le cycle de vie sera une combinaison adaptée des phases canoniques. Le cycle de vie canonique ne décrit pas les processus de support de projet tels le processus de management, l'assurance qualité, la logistique.

L'étude d'opportunité permet de définir et d'évaluer les objectifs tangibles et intangibles du projet :

-> Définir les objectifs, le périmètre et les grandes lignes de la solution
-> Vérifier qu'ils sont alignés avec la stratégie de l'entreprise
-> Identifier les risques et les moyens de les contenir
-> Définir les gains attendus
-> Définir les coûts maximaux

L'étude de faisabilité permet d'établir la faisabilité du projet, c'est-à-dire d'établir s'il existe une solution technique optimale répondant aux exigences techno-économiques identifiées lors de la phase précédente :

-> Établir l'expression des besoins
-> Identifier les solutions possibles
-> Identifier les risques techniques, financiers et projet
-> Préconiser la solution optimale
-> Faire une estimation détaillée du projet
-> Faire un planning
-> Rédiger un cahier des charges

La phase de conception définit le processus de réalisation. Elle comprend :

-> La conception de l'architecture générale du système
-> La conception de l'architecture détaillée
-> La conception détaillée de chaque composant (interfaces, services, base de données)
-> La conception des plans de tests

La phase de réalisation est la phase où est produit l'objet du projet. Elle comprend :

-> La programmation des composants
-> Les tests unitaires
-> La réalisation des jeux d'essai
-> Les tests d'interface
-> Les tests d'intégration (reprises, performance...)

La phase de recette est la phase de contrôle qualité, où l'on s'assure que le produit fini répond bien à la demande initiale, à l'intérieur des contraintes posées :

-> Les tests de validation
-> Les tests en parallèle
-> Les tests de non régression
-> Les tests de réception ou recette

La phase de mise en production s'achève par la mise en production.

-> Les tests de mise en production

La phase de maintenance est la phase qui permet à l'objet du projet de se maintenir au cours du temps.

-> La gestion du changement et des évolutions

Il existe plusieurs types de modèles :

-> Les modèles de cycle de vie linéaire

- Modèle en cascade
- Modèle en V
- Modèle en W

Chaque phase du cycle de vie doit être réalisée avec tous les détails requis avant de passer à la phase suivante. Il y a en particulier un effort important à fournir pour la documentation.

Il n'existe pas de version du logiciel exécutable avant la fin du développement. Toute reprise, en cas d'écart entre la compréhension des besoins et le besoin réel, est coûteuse.

-> Les modèles de cycle de vie non linéaire

Les logiciels réalisés suivant ces modèles sont complétés par itérations ou incréments successifs. Les détails de réalisation peuvent être reportés afin de produire une version opérationnelle du logiciel au plus tôt. Ce type de modèles semble plus approprié pour prendre en compte le caractère évolutif des besoins des utilisateurs.

- Modèle incrémental
- Modèle évolutif
- Modèle en spirale
- Modèle de cycle RAD
- …

Il faut souligner la différence entre étapes (découpage temporel) et activités (découpage selon la nature du travail). Il y a des activités qui se déroulent dans plusieurs étapes (ex : La spécification, la validation et la vérification), voire dans toutes les étapes (ex : La documentation).

MODELE EN CASCADE

Dans ce modèle, les étapes sont réalisées de façon séquentielle. Chaque étape donne lieu à l'établissement d'un document.

Analyse des besoins ou analyse préalable :

Document final : Cahier des charges + plan qualité

-> Qualités fonctionnelles attendues en termes des services offerts,
-> Qualités non fonctionnelles attendues : Efficacité, sûreté, sécurité, facilité d’utilisation, portabilité...
-> Qualités attendues du procédé de développement (ex : Procédures de contrôle qualité).

Le cahier des charges peut inclure une partie destinée aux clients (définition de ce que peuvent attendre les clients) et une partie destinée aux concepteurs (spécification des besoins).

Étude préliminaire ou étude de faisabilité ou planification :

Document final : Rapport d’analyse préliminaire ou schéma directeur

-> Définition globale du problème,
-> Différentes stratégies possibles avec avantages/inconvénients,
-> Ressources, coûts, délais.

Analyse du système :

Analyse détaillée de toutes les fonctions et autres caractéristiques que le logiciel devra réaliser pour l'usager, tel que vu par l'usager.

Résultat : Document de spécifications qui définit le « quoi » du logiciel. Ce document doit être rédigé dans un langage aussi formel que possible.

-> Modélisation du domaine,
-> Modélisation de l'existant (éventuellement),
-> Définition d'un modèle conceptuel (ou spécification conceptuelle),
-> Plan de validation.

Conception générale :

Définition de l'architecture générale du logiciel. Pas de détails sur la manière dont les éléments composant le système seront implantés.

-> Résultat : Document de conception générale

Conception détaillée :

Spécification de la manière dont chacun des composants du logiciel sera réalisé et dont ils interagiront.

-> Résultat : Documents de conception détaillée.

Codage et tests unitaires :

Document final : Dossiers de programmation et codes sources

-> Traduction dans un langage de programmation,
-> Tests avec les jeux d'essais par module selon le plan de test.

Intégration et tests de qualification :

-> Composition progressive des modules,
-> Tests des regroupements de modules,
-> Test en vraie grandeur du système complet selon le plan de test global (« alpha testing »).

Installation :

Mise en fonctionnement opérationnel chez les utilisateurs. Parfois restreint dans un premier temps à des utilisateurs sélectionnés (« beta testing »).

Maintenance :

-> Maintenance corrective (ou curative),
-> Maintenance adaptative,
-> Maintenance perfective.

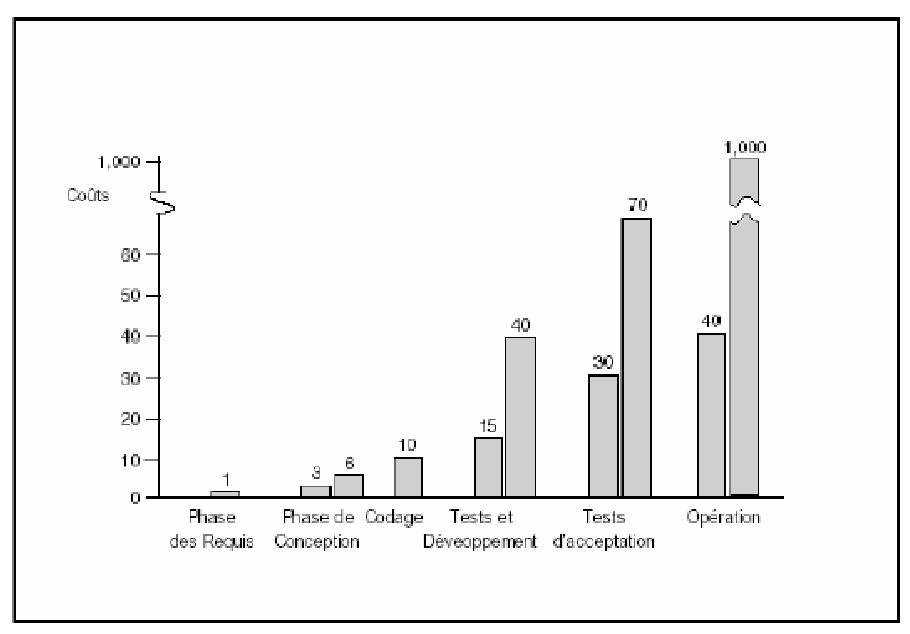
Activités transversales à tout le cycle de vie :

-> Spécification, documentation, validation et vérification, management.

Des études ont été menées pour évaluer le coût des différentes étapes du développement:

Type de système	Conception	Implantation	Test
Gestion	44 %	28 %	28%
Scientifique	44 %	26 %	30%
Industriel	46 %	20 %	34%

Mais c'est la maintenance qui coûte le plus cher.



Coûts relatifs de correction des exigences, lorsque découvertes à différents stades

Les avantages du modèle en cascade

-> On a une idée claire de ce qu'il y a à faire.
-> Il est simple à comprendre.
-> Il permet une normalisation des cadres conceptuels et terminologiques des différentes activités.

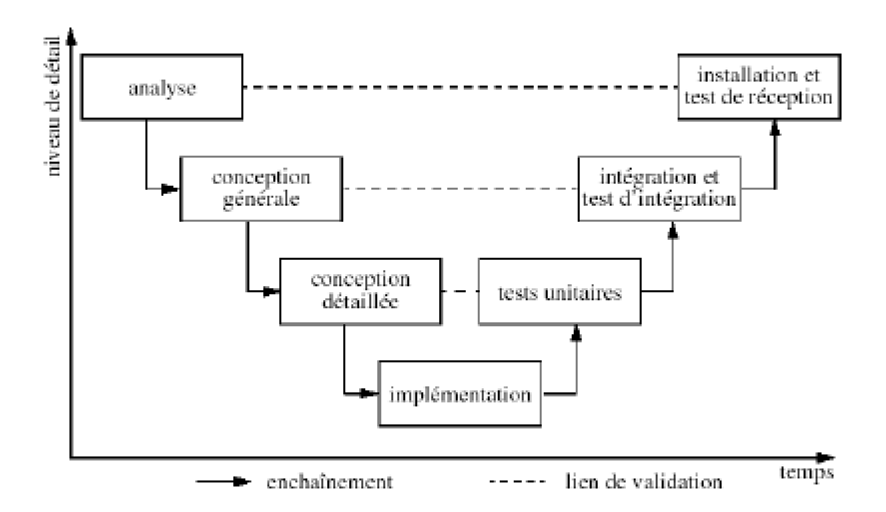
Inconvénients :

-> Il ne reflète pas la façon dont le code est réellement développé.
-> Il manque de flexibilité en cas d'imprévus.
-> Il n'y a pas de retour avant la livraison chez le client.

MODELE EN V

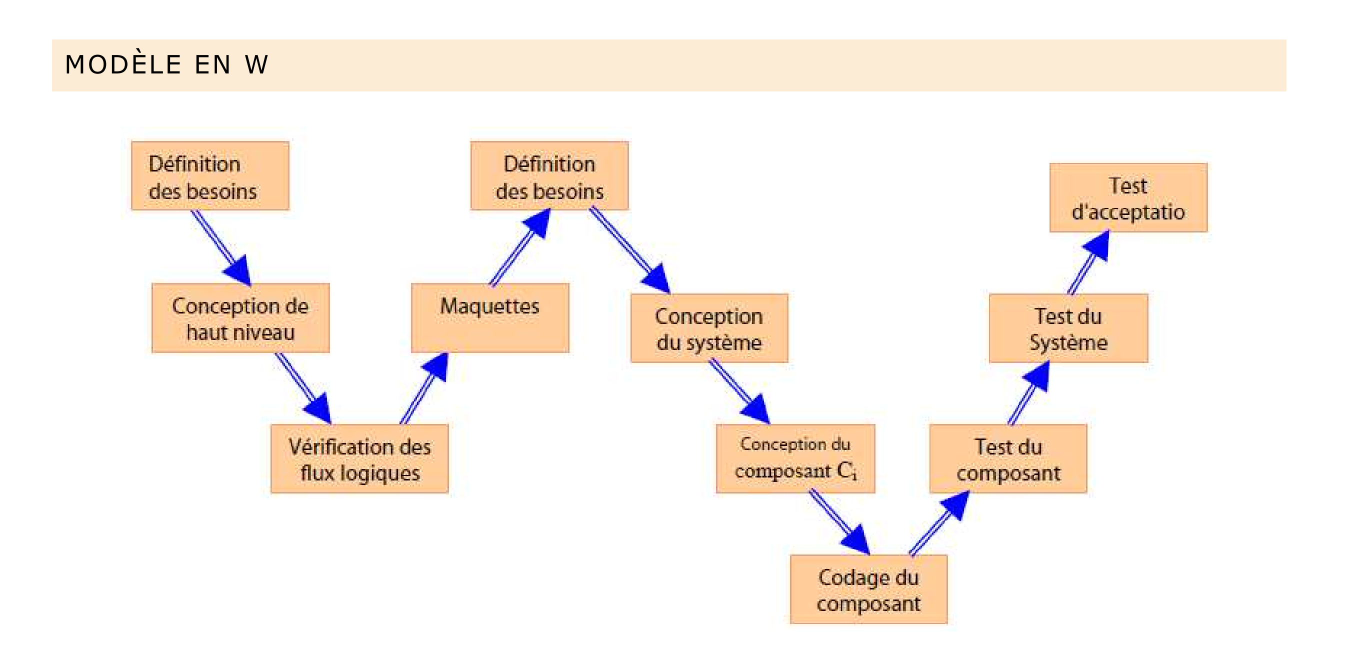
Le modèle en V est une autre façon de présenter une démarche qui reste linéaire, mais qui fait mieux apparaître les produits intermédiaires à des niveaux d'abstraction et de formalité différents et les procédures d'acceptation (validation et vérification) de ces produits intermédiaires.

Le V est parcouru de gauche à droite en suivant la forme de la lettre : les activités de construction précèdent les activités de validation et vérification. Mais l'acceptation est préparée dès la construction (flèche de gauche à droite). Cela permet de mieux approfondir la construction et de mieux planifier la 'remontée' d'abstraction et de formalité différentes et les procédures d'acceptation (validation et vérification) de ces produits intermédiaires.



Il a pour avantage de montrer comment les activités de "tests" sont liées à celles d'analyse et de conception. Comme pour le modèle en cascade, il y a un manque de retour, pas de résultats intermédiaires servant de base à la discussion avec le client.

MODÈLE EN W



MODELE INCREMENTAL

-> Division du logiciel en sous-systèmes (un par fonctionnalité)
-> Première version : Logiciel partiel
-> Chaque nouvelle version ajoute un nouveau sous-système

Ce modèle consiste à développer le futur logiciel par lots et à établir un planning de réalisation de ces lots avant le démarrage du processus de développement. Chaque lot doit fournir un produit qui sera immédiatement mis en exploitation dans un environnement opérationnel.

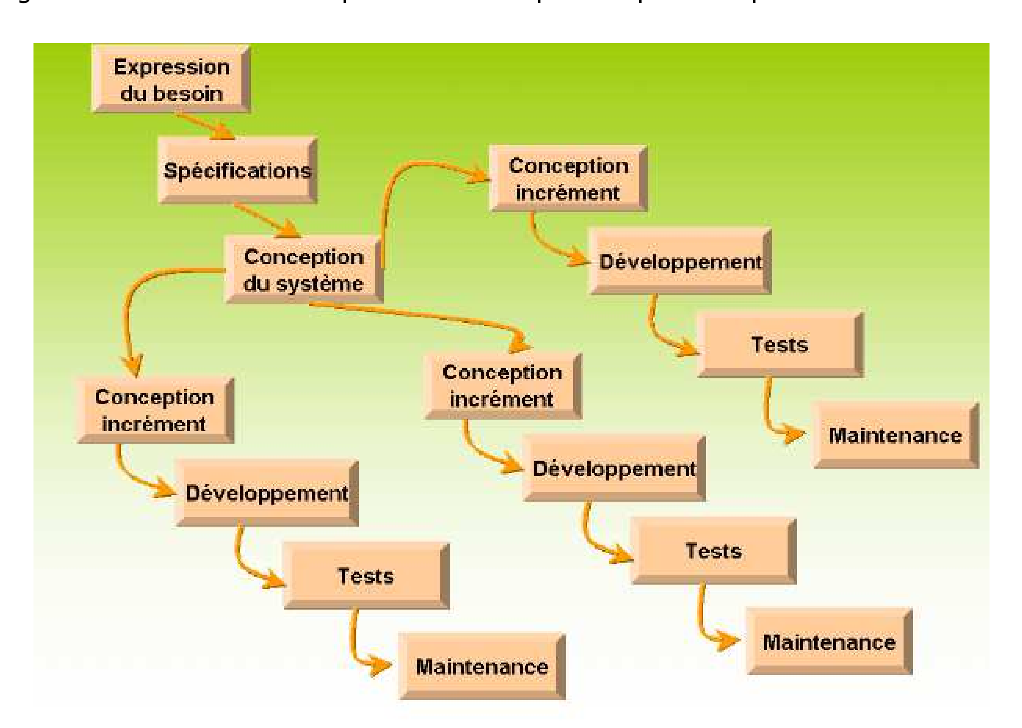
L'ensemble des produits en exploitation à un moment donné constitue une version provisoire du logiciel dont la durée de vie s'étend jusqu'à la livraison du prochain lot.

Les avantages d'un tel modèle :

-> Les efforts de maintenance d'une version provisoire sont faibles.
-> Chaque nouvelle version améliorant la précédente, il y a une meilleure adéquation entre les besoins des utilisateurs et les besoins couverts par le logiciel.

Les inconvénients :

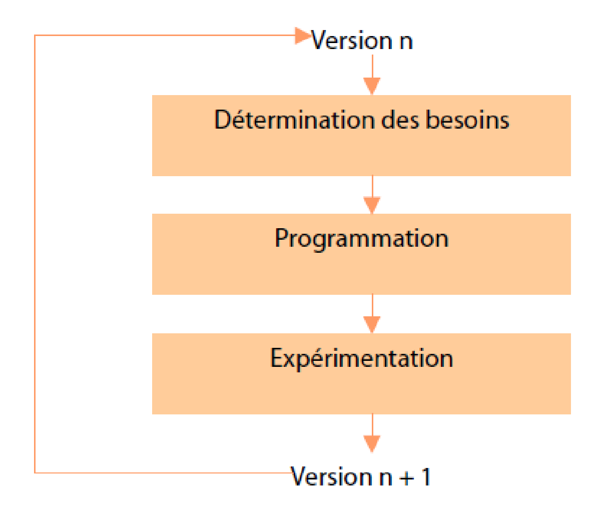
-> Il faut spécifier dès le début du développement l'architecture globale du logiciel et définir les lots qui seront développés : le noyau, les incréments qui doivent être fonctionnellement indépendants, leur intégration.
-> L'intégration de nouveaux composants est de plus en plus complexe.



C'est un modèle adapté aux grands projets. Néanmoins, l'architecture du système doit permettre de définir des domaines suffisamment découplés. Dans le cas contraire, certains incréments doivent attendre que les incréments avec lesquels ils sont liés soient suffisamment développés. Lorsqu'on leur propose un développement par lot, les Maîtrises d'Ouvrage doivent vérifier le couplage des domaines.

MODELE EVOLUTIF

-> Approche itérative
-> Division du logiciel en sous-systèmes (un par fonctionnalité)
-> Première version : Coquille complète du logiciel
-> Chaque nouvelle version apporte une modification / amélioration à un sous-système



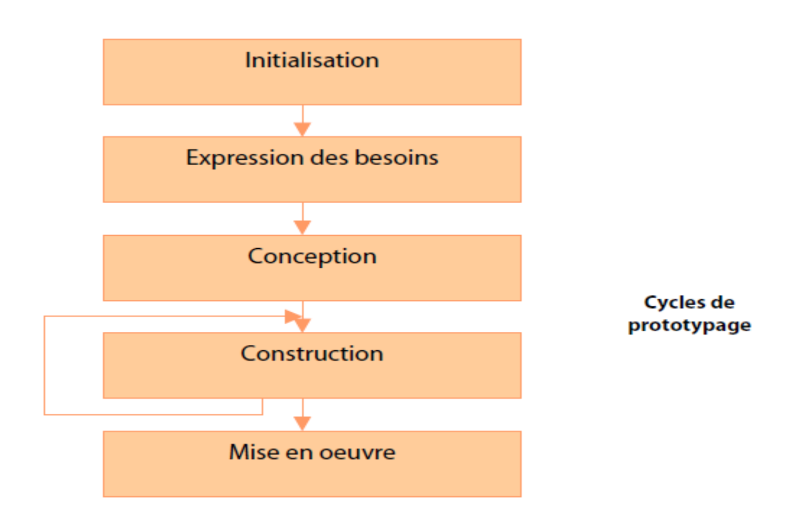
Avantages

-> Formation précoce des utilisateurs, retours rapides.
-> Création précoce de nouveaux marchés pour les nouvelles fonctionnalités.
-> Focalisation sur le nouveau domaine d'expertise à chaque étape (version).
-> Détection précoce des problèmes imprévus (correction immédiate du système en développement).

Inconvénients

-> Risque de la remise en cause du noyau (fonctionnalités de base) au cours du développement.

MODÈLE DE CYCLE RAD



Structure d'une phase dans le cycle RAD

Travaux préparatoires	Session participative	Travaux de conclusion

Approche de construction de systèmes d'information qui combine l'utilisation d'outils **CASE**, du prototypage itératif et de l'observation d'échéanciers rigoureux dans une méthodologie efficace visant à réduire les coûts de développement tout en assurant une qualité raisonnable du produit.

Constat fréquent

-> Une solution qui satisfait 4/5 des exigences peut être produite dans le ¼ du temps requis pour développer une solution complète.
-> Les exigences de rentabilité d'un logiciel peuvent être entièrement satisfaites même si certaines exigences fonctionnelles ne le sont pas encore tout à fait.
-> L'acceptabilité d'un logiciel peut être déterminée sur la base convenue sous-ensemble minimum d'exigences utiles plutôt que sur toutes les exigences définies.

Objectifs

-> Réduire le temps de développement (quitte à faire certains compromis sur les coûts et les exigences de qualité).
-> Permettre au client de juger le plus tôt possible.
-> Mise en marché rapide d'un sous-ensemble utile du logiciel.
-> Pour satisfaire les exigences de rentabilité exprimées par le client.
-> Pour limiter les menaces de changements tardifs auxquelles le projet s'expose N itérations

Une question de compromis.

-> Temps / coûts / qualité.

Les compromis déterminent le rythme du développement

-> Développement efficace : Équilibre entre coût, temps et qualité.

● RAD normal : Compromit au niveau des coûts et de la qualité pour favoriser la réduction du temps de développement.
● RAD extrême : Codage intensif!! (Coûts plus élevés, qualité moindre).

Contraintes du processus

-> Le projet ne doit pas durer plus de 6 mois.
-> Le développement itératif doit converger vers une solution rentable et acceptable pour l'entreprise.
-> Pour être accepté, chaque livrable (version du logiciel, documents, etc.) doit répondre à un besoin de l'entreprise.
-> Toute partie pouvant influencer la définition des exigences doit être représentée dans l'équipe de développement tout au long du processus.
-> Les clients, les développeurs et gestionnaires doivent accepter des documents informels

-> L'équipe de développement doit avoir l'autorisation de prendre des décisions généralement réservées aux gestionnaires.

Caractéristiques

-> Équipes hybrides.
-> Environ six personnes incluant développeurs, utilisateurs, personnes impliquées dans la définition des exigences du projet (client).
-> Les développeurs choisis devront être flexibles et polyvalents de façon à pouvoir assumer à la fois les rôles d'analyse, de concepteurs et de programmeurs.
-> Utilisation d'outils spécialisés supportant.
-> Développement visuel, création de prototypes (jetables et incrémentales), langages multiples, planification, travail d'équipe et en collaboration, utilisation de composants réutilisables, utilisation de bibliothèques, gestion de versions.
-> Encadrement strict du temps (timeboxing).
-> Élimination des aspects secondaires (définitivement ou temporairement) afin de respecter des échéanciers fixes.

Développement par prototypage itératif et incrémental

À chaque itération :

-> Développement d'une version du logiciel.
-> La durée de chaque itération varie entre une journée et trois semaines.
-> Session de groupe (focus group) pour la mise au point.
-> Environ 2 heures.
-> Évaluation de la dernière version.
-> Planification de la prochaine version : on priorise les exigences et on élimine celles qui ne pourront être réalisées dans le temps prévu alloué.

Le RAD n'est pas approprié :

-> Pour les applications nécessitant d'interagir avec d'autres.
-> Pour des applications critiques.
-> Lorsqu'une performance optimale est requise.
-> Lorsqu'une grande fiabilité est exigée.
-> Lorsque la technologie requise n'est pas bien maîtrisée.

MODELE EN SPIRALE

Enfin le modèle en spirale, de Boehm (88), met l'accent sur l'évaluation des risques. À chaque étape, après avoir défini les objectifs et les alternatives, celles-ci sont évaluées par différentes techniques (prototypage, simulation...), l'étape est réalisée et la suite est planifiée. Le nombre de cycles est variable selon que le développement est classique ou incrémental.

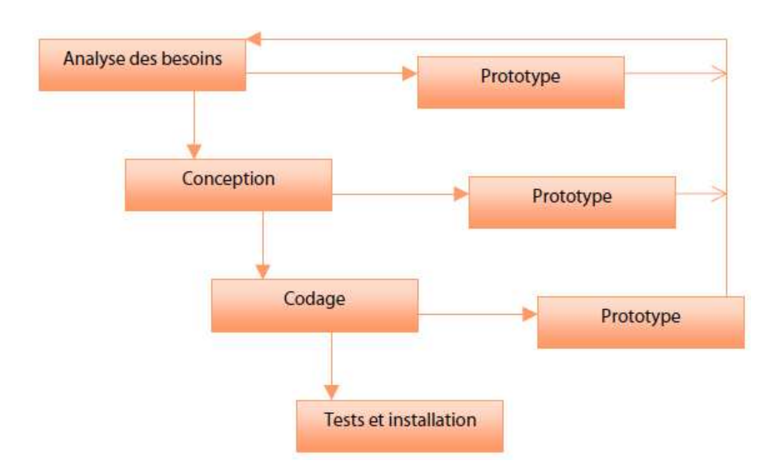
La démarche à suivre est :

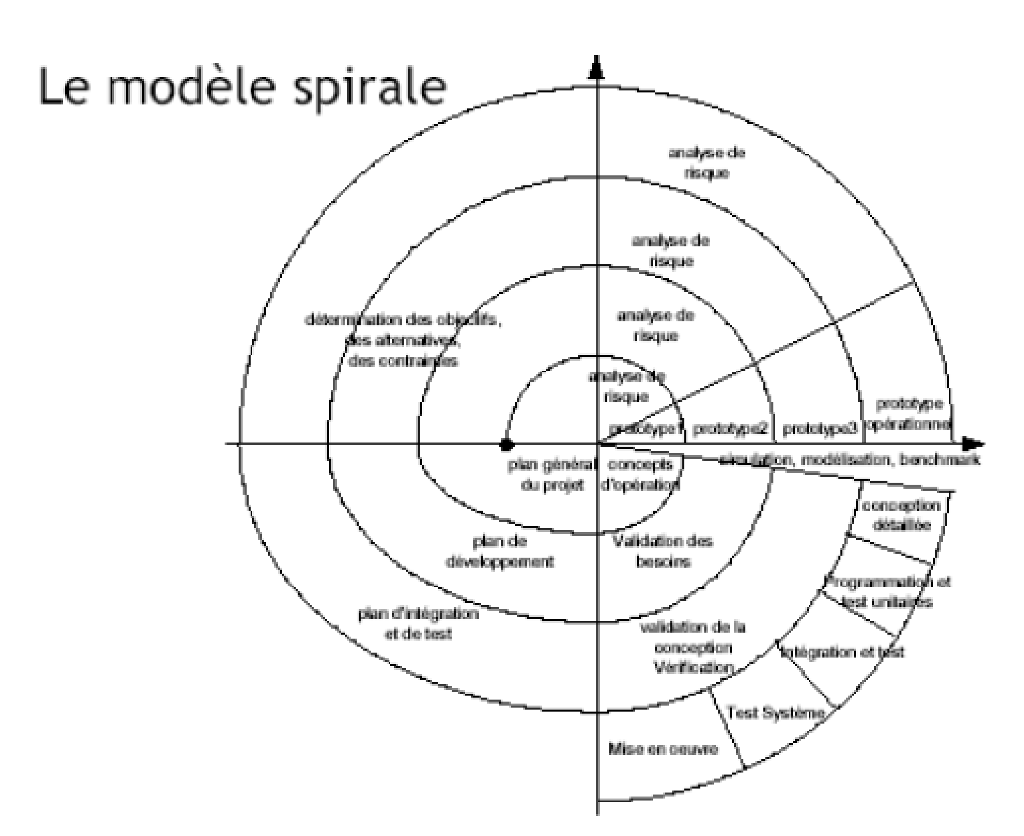
-> Développement de sous-ensembles représentatifs du logiciel
-> Validation des composantes en fonction des besoins fonctionnels, des contraintes matérielles/logicielles…

Le déroulement à suivre est : composé d'une suite de cycles de développement de 4 phases dont le nombre n'est pas déterminé à l'avance et qui dépend de l'approche de développement choisie pour réaliser le produit issu de chaque phase ou cycle.

Chaque cycle de la spirale se déroule en 4 phases :

-> Phase d'identification : Identification des besoins, détermination des objectifs, des alternatives pour les atteindre et des contraintes.
-> Phase d'évaluation : Analyse des risques, évaluation des alternatives, identifier et résoudre les risques.
-> Phase de réalisation : Développement et vérification de la solution à l'issue de la phase précédente.
-> Phase de vérification et validation du produit élaboré dans la phase précédente, planifier la prochaine phase.





Points forts :

-> Produire rapidement un logiciel opérationnel minimal. Permets de se concentrer sur les aspects les plus incertains du développement.
-> Suppose une discipline stricte pour éviter de « coder/finaliser ». Il faut accepter les remises en cause de la part du client à chaque nouvelle évaluation.
-> Son ouverture à l'ensemble des approches et technologies de développement existantes
-> son caractère itératif. L'intégration d'une approche orientée – risque : Les risques peuvent être appréhendés dès le début du projet, lors de la réalisation/livraison de la première version/composante du logiciel et permettre de réviser les choix effectués au cours des différents cycles en fonction de l'avancement du projet.
-> Évite d'approfondir la spécification trop tôt. Seules les spécifications de la première version/composante sont définies, les autres le seront plus tard.

Points faibles :

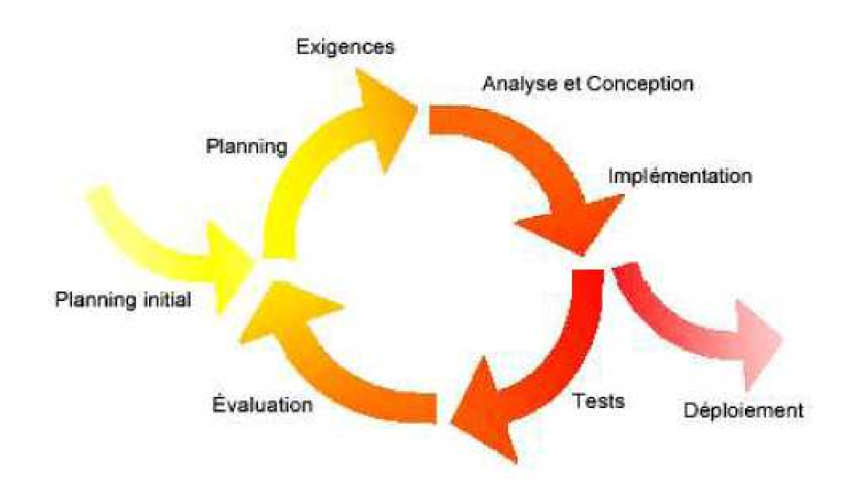
-> Difficultés pour mener à bien les premiers cycles de la spirale.
-> Risque de remise en cause des spécifications des modules/versions déjà réalisé(e)s lors de l'analyse de nouveaux modules/versions.
-> Difficultés de contrôle du processus.
-> Organisation du développement souvent modifiée pour le client.

UP

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

UP est itératif

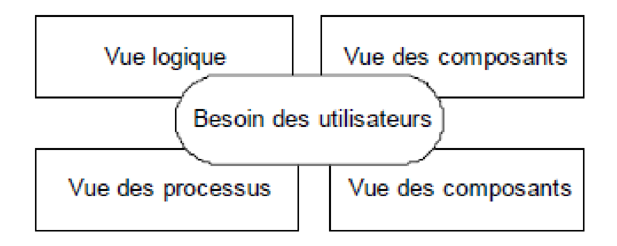
L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes. Elle qualifie un traitement ou une procédure qui exécute un groupe d'opérations de façon répétitive jusqu'à ce qu'une condition bien définie soit remplie.



Une itération prend en compte un certain nombre de cas d'utilisation et traite en priorité les risques majeurs.

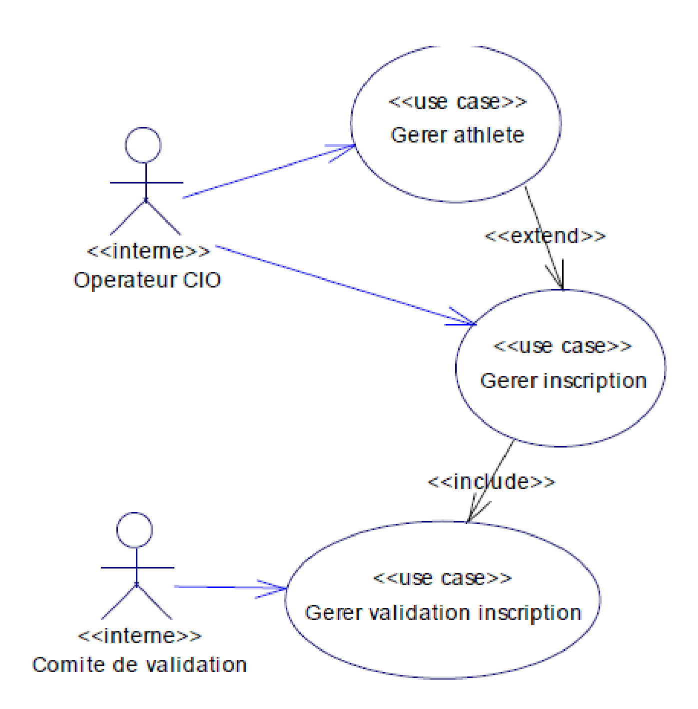
UP est centré sur l'architecture

Ph. Kruchten propose différentes perspectives, indépendantes et complémentaires, qui permettent de définir un modèle d'architecture (publication IEEE, 1995).



UP est piloté par les cas d'utilisation d'UML

Le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc accès sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.



Vie du processus unifié

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. UP est un ensemble de principes génériques adapté en fonction des spécificités des projets. UP répond aux préoccupations suivantes :

- **QUI** participe au projet ?
- **QUOI**, qu'est-ce qui est produit durant le projet ?

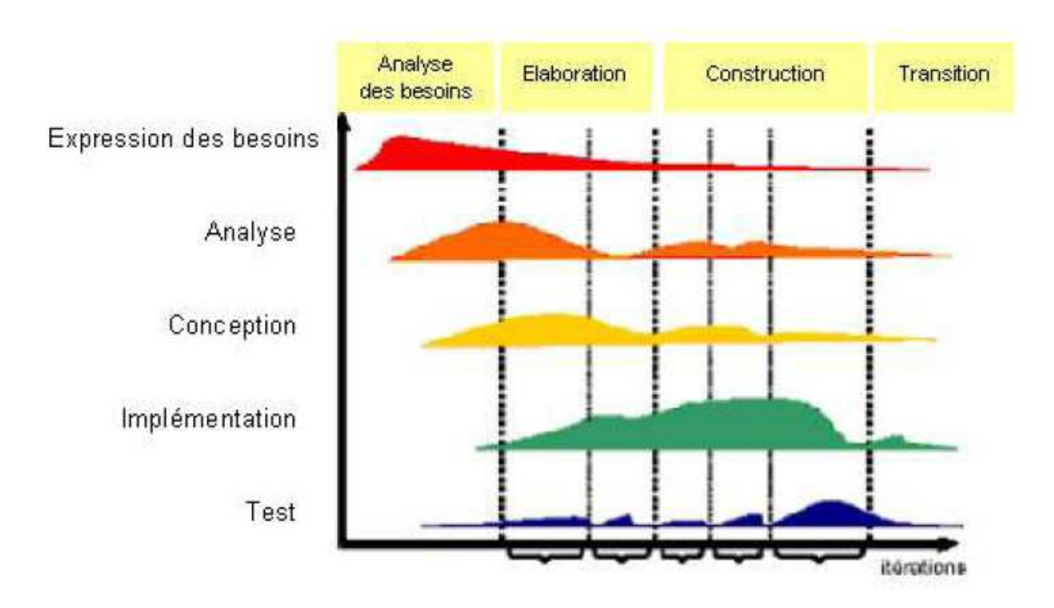
- **COMMENT** doit-il être réalisé ?
- **QUAND** est réalisé chaque livrable ?

L'architecture bidirectionnelle

UP gère le processus de développement par deux axes.

L'axe vertical représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en terme décomposant, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus. Cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en termes de cycles, de phases, d'itérations et de jalons.



UP répète un certain nombre de fois une série de cycle qui s'articule autour de 4 phases :

- Analyse des besoins
- Élaboration
- Construction
- Transition

Pour mener efficacement un tel cycle, les développeurs ont besoin de toutes les représentations du produit logiciel.

- Un modèle de cas d'utilisation.
- Un modèle d'analyse : Détailler les cas d'utilisation et procéder à une première répartition du comportement.
- Un modèle de conception : Finissant la structure statique du système sous forme de sous-systèmes, de classes et interfaces.
- Un modèle d'implémentation : Intégrant les composants.
- Un modèle de déploiement : Définissant les nœuds physiques des ordinateurs.
- Un modèle de test : Décrivant les cas de test vérifiant les cas d'utilisation.
- Une représentation de l'architecture.

Les activités

__Expression des besoins__ : L'expression des besoins comme son nom l'indiquent, permettent de définir les différents besoins :

- Inventorier les besoins principaux et fournir une liste de leurs fonctions.
- Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation.
- Appréhender les besoins non fonctionnels (technique) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

__Analyse__ : L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilitent la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

__Conception__ : La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation.

Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation :

- Elle décompose le travail d'implémentation en sous-système.
- Elle crée une abstraction transparente de l'implémentation.

__Implémentation__ : L'implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous forme de codes sources.

__Test__ : Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

Les phases

__Analyse des besoins__ : L'analyse des besoins donne une vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs, les délais et les coûts.

On met en place le projet. Elle répond aux questions suivantes :

- Que va faire le système ? Par rapport aux utilisateurs principaux, quels services va-t-il render ?
- Quelle va être l'architecture générale (cible) de ce système
- Qu'elles vont être : les délais, les coûts, les ressources, les moyens à déployer ?

Élaboration : L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. L'élaboration permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence. Au terme de cette phase, les chefs de projet doivent être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet. Les tâches à effectuer dans la phase élaboration sont les suivantes :

- Créer une architecture de référence.
- Identifier les risques, ceux qui sont de nature à bouleverser le plan, le coût et le calendrier
- Définir les niveaux de qualité à atteindre.
- Formuler les cas d'utilisation pour couvrir les besoins fonctionnels et planifier la phase de construction.
- Élaborer une offre abordant les questions de calendrier, de personnel et de budget.

Construction : La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d'utilisation que les chefs de projet en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

Transition : Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

RUP

"Rational Unified Process" est une méthode développement propriété de la société Rational et basé sur l'utilisation de l'atelier Rational. Elle est vendue sous la forme d'une documentation hypertexte et d'outils incorporables à chaque composant de l'AGL de Rational. C'est une méthode évolutive, qui met en œuvre les mêmes principes que le RAD ou le développement en spirale.

Elle est composée de 4 phases :

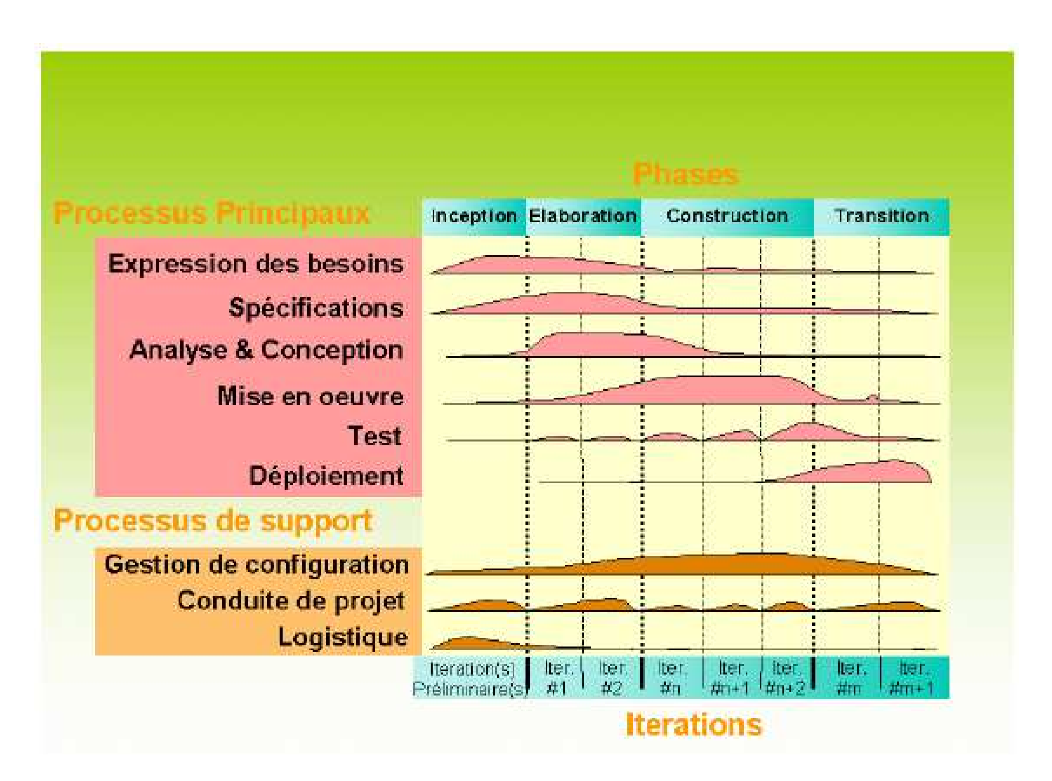
-> Inception : Initial "use case", coût/bénéfice, risques, planification
-> Élaboration : "use case" à 80% terminés, architecture logicielle, prototype
-> Construction : Développement
-> Transition : Tests, conversion et déploiement

RUP utilise le niveau élevé d'automatisation que lui procure l'AGL Rational. Il permet :

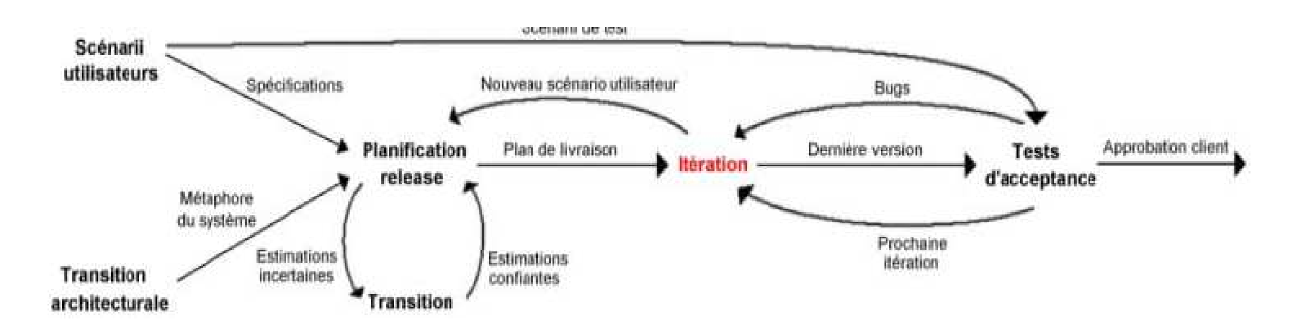
-> D'une part de planifier des itérations à l'intérieur de chaque phase,

-> De paralléliser les processus au maximum : ainsi la définition d'architecture peut commencer tandis que les spécifications ne sont pas terminées.

Le résultat peut être impressionnant, si l'équipe projet est composée d'experts, maîtrisant parfaitement les risques projet. C'est probablement une voie très intéressante, lorsque des standards multi-fournisseurs seront adoptés. En effet, les maîtrises d'ouvrage devront être attentives à ne pas s'enfermer dans méthodes qui lient trop la compétence de développement des systèmes d'information au savoir-faire d'un fournisseur.



LE CYCLE DE L'EXTREME PROGRAMMING



Une méthode agile

Comme toutes les méthodes de développement, l'Extreme Programming propose un cadre pour l'ensemble des aspects du projet logiciel, depuis l'analyse des besoins jusqu'aux tests, en passant par la conception. Mais à la différence des processus prédictifs, recourant généralement à UML (même si XP utilise aussi parfois UML comme support de communication), XP ne se fonde pas sur la définition exhaustive et précoce des besoins ; elle parie plutôt, à partir d'un ensemble de règles strictes, sur la souplesse et la mise en valeur du "capital humain".

La programmation comme discipline collective

Tout en mettant l'accent sur les bonnes pratiques de programmation, XP préconise un déroulement par itération courte et gérée collectivement, avec une implication constante du client. Il en découle une redéfinition de la relation entre client et fournisseur, avec de surprenants résultats en termes de qualité de code, de délais et de satisfaction de la demande du client.

Valeurs

L'Extreme Programming repose sur 4 valeurs fondamentales :

-> La communication : C'est le moyen fondamental d'éviter les erreurs. Le moyen à privilégier est la conversation directe, en face à face. Les moyens écrits ne sont que des supports et des moyens de mémorisation.
-> Le courage : Il est nécessaire à tous les niveaux et de la part de tous les intervenants, notamment chez les développeurs (quand des changements surviennent à un stade avancé du projet, ou quand des défauts apparaissent) et chez le client (qui doit pouvoir prioriser ses demandes).
-> Le retour d'information (feedback) : Les itérations sont basées sur les retours d'informations du client, permettant une adéquation totale entre l'application finale et sa demande.
-> La simplicité : En Extreme Programming, la façon la plus simple d'arriver à un résultat est la meilleure. Prévoir préalablement des évolutions futures n'a pas de sens. Ce principe est résumé en une phrase : "Tu n'en auras pas besoin." (en anglais « You ain't gonna need it »). La meilleure manière de rendre une application extensible est de la rendre aussi simple (et donc simple à comprendre) et bien conçue que possible.

Pratiques

Ces 4 valeurs se déclinent en 13 pratiques :

-> **Un représentant du client sur site :** Afin d'assurer une meilleure réactivité et un feedback rapide, un représentant du client doit être présent pendant toute la durée du projet. Ce représentant doit avoir les connaissances d'un utilisateur final, mais en même temps une vision globale du résultat à obtenir.

-> **Planning game** : Le client créé des scénarii d'utilisation, en les priorisant. Ces scénarii seront ensuite implémentés par l'équipe de développement. Le projet est considéré comme achevé quand le client n'est plus en mesure de trouver de nouveau scénario.

-> **Intégration continue** : L'intégralité de ce qui est développé est assemblée à chaque fois qu'une tâche est terminée, ce qui permet d'avoir toujours une version à jour, notamment comme livrable ou pour le démarrage de nouvelles tâches.

-> **Release fréquente** : Les livraisons doivent être les plus fréquentes possible, afin d'obtenir un feed-back le plus rapidement possible, tout en offrant des fonctionnalités complètes. La fréquence de livraison est donc de quelques semaines

-> **Rythme soutenables** : Afin d'éviter de surcharger l'équipe, elle ne fait pas d'heures supplémentaires deux semaines de suite. Si le cas se présentait, cela signifierait qu'il faut redéfinir le planning.

-> **Tests de recette** : À partir de critères définis par le client, on crée des procédures de test, souvent automatisées, qui permettent de vérifier fréquemment le bon fonctionnement de l'application.

-> **Tests unitaires** : Avant d'implémenter une fonctionnalité, il convient d'écrire un test qui validera l'implémentation. Ces tests sont issus de la traduction à plus bas niveau des Tests de recette. Il est donc normal qu'XP pousse la notion de Tests unitaires (Unit Tests) en imposant leur écriture avant celui du code (first-test).

-> **Conception simple :** L'objectif est de produire une application qui réponde aux attentes du client. Mieux vaut donc arriver à ce résultat de la manière la plus simple possible, afin de faciliter les évolutions futures en rendant le compte simple à comprendre et facile à modifier. De même, la documentation doit être minimale, c'est-à-dire ce qui est demandé par le client.

-> **Utilisation de métaphores** : On utilise des métaphores et des analogies pour décrire le système et son fonctionnement, ce qui permet de le rendre compréhensible par les non-informaticiens, comme les utilisateurs ou les commerciaux.

-> **Refactoring du code** : Amélioration continue de la qualité du code sans en modifier le comportement (commentaire du code, respect des règles de nommage, simplification…).

-> **Convention de nommage** : Dans l'optique d'appropriation collective du code et de facilitation de la communication, il est indispensable d'établir et de respecter des normes de nommage pour les variables, méthodes, objets, classes, fichiers…

-> **Programmation en binôme** : La programmation se fait par deux. Le premier, appelé driver, a le clavier. C'est lui qui va travailler sur la portion de code à écrire. Le second, appelé partner, est là pour l'aider, en suggérant de nouvelles possibilités ou en décelant d'éventuels problèmes. Les développeurs changent fréquemment de partenaires, ce qui permet d'améliorer la connaissance collective de l'application et d'améliorer la communication au sein de l'équipe.

-> **Appropriation collective du code** : L'équipe est collectivement responsable de l'application et est supposée connaître l'intégralité du code. Chacun peut également faire des modifications dans toutes les portions du code, même celles qu'il n'a pas écrites.

Autres principes :

-> Ne pas ajouter de fonctionnalités plus tôt que prévu.
-> N'optimiser qu'à la toute fin.

Cette méthode s'appuie sur :

- Une forte réactivité au changement des besoins du client.
- Un travail d'équipe.
- La qualité du code.
- La qualité des tests effectués au plus tôt.

Une méthode qui ne fait pas l'unanimité

L'Extreme Programming est parfois critiqué pour sa radicalité. En effet, si on en croit les chantres de cette méthode, et notamment ses auteurs, on ne fait de l'Extreme Programming que si l'on met en place toutes les pratiques de la méthode, ce que certains trouvent trop contraignant et contraire à l'esprit agile, qui doit justement permettre d'assouplir l'aspect méthodologique des projets informatiques.

On reproche aussi parfois à cette méthode d'être mal adaptée à l'environnement économique : comment prévoir la durée et le coût d'un projet ?

Il n'y a pas de modèle idéal, car tout dépend des circonstances. Le modèle en cascade ou en V est risqué pour les développements innovants, car les spécifications et la conception risquent d'être inadéquates et souvent remises en cause. Le modèle incrémental est risqué, car il ne donne pas beaucoup de visibilité sur le processus complet. Le modèle en spirale est un canevas plus général qui inclut l'évaluation des risques.

Souvent, un même projet peut mêler différentes approches, comme le prototypage pour les sous-systèmes à haut risque et la cascade pour les sous-systèmes bien connus et à faible risque.

REPONDRE A UN APPEL D'OFFRE DECISIONNEL

APPEL D'OFFRE DECISIONNEL

Répondre à des appels d'offres est souvent nécessaire pour la réalisation de projets au forfait. L'art est difficile et le client jugera de la qualité de votre réponse en fonction de critères qui lui seront propres. Bien entendu, c'est lors de la réunion que vous aurez sollicité qu'il vous les expliquera et vous en dira certainement plus.

Avoir un entretien téléphonique à défaut d'un rendez-vous, vous aidera à répondre, mais le contact humain est à privilégier au maximum sous peine de partir avec un handicap certain.

Si vous connaissez le client ou êtes déjà chez lui, tout sera vraiment plus facile

Note 1 : Dans le cadre de marché public, le RDV est généralement plus difficile à avoir, parfois impossible.

Note 2 : Demandez toujours la date de démarrage prévue ainsi que le planning de réalisation si cette partie est floue, ce sera un indicateur de charge et vous pourrez alors aborder la question du budget avec une question directe ou à défaut si le montant auquel vous pensez n'est pas exagéré.

Vous serez généralement jugé sur 4, 5, 6 critères en moyenne comme par exemple :

- La compréhension de la demande et du contexte.
- Une réponse claire et cohérente.
- Le savoir-faire technique et/ou fonctionnel.
- Des expériences similaires.
- La disponibilité des ressources et des compétences.
- La qualité des CV.
- Un planning cohérent.
- Des charges équilibrées.
- Un prix compétitif.
- Souvent et surtout un prix compétitif.
- ...

LA REPONSE

Qui n'a pas entendu, tu as une réponse à produire pour tel jour, vas-y. Première chose à faire, fouiller dans ses archives, le réseau ou demander à des collègues pour trouver une réponse similaire et gagner du temps. Évidemment, si l'on est à plusieurs, c'est plus facile.

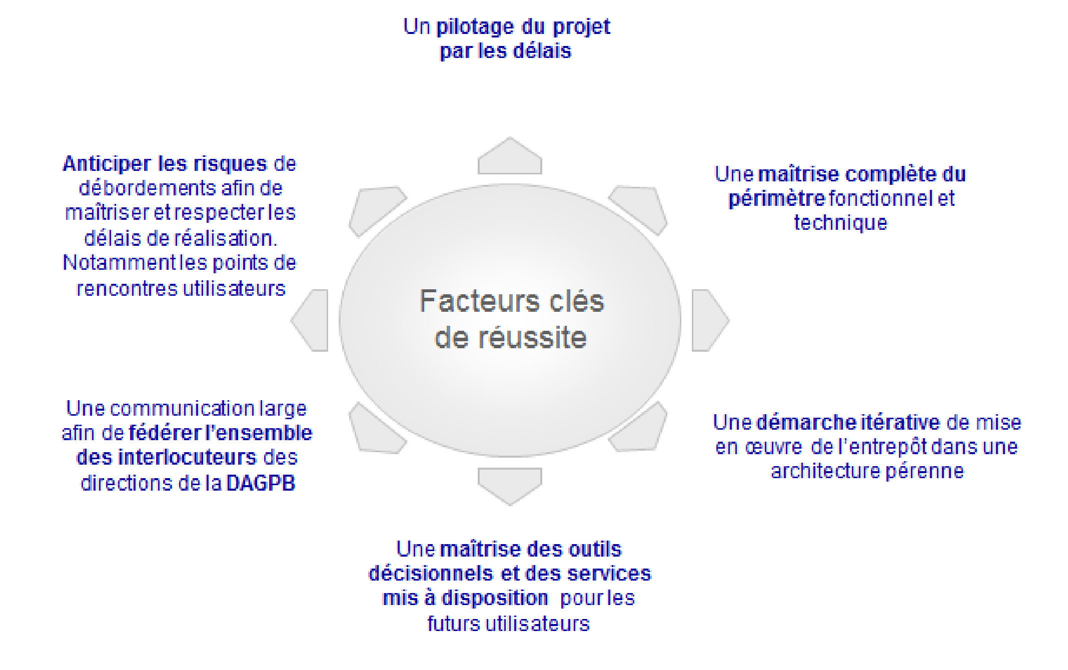
Si vous êtes seul et que vous n'avez rien ou peu, n'êtes pas inspiré, c'est votre ou vos première(s) fois, ou tout simplement c'est toujours difficile, alors, dans la mesure de vos moyens, demander aux « autres » de vous fournir des éléments de réponse.

FACTEURS CLES DU SUCCES

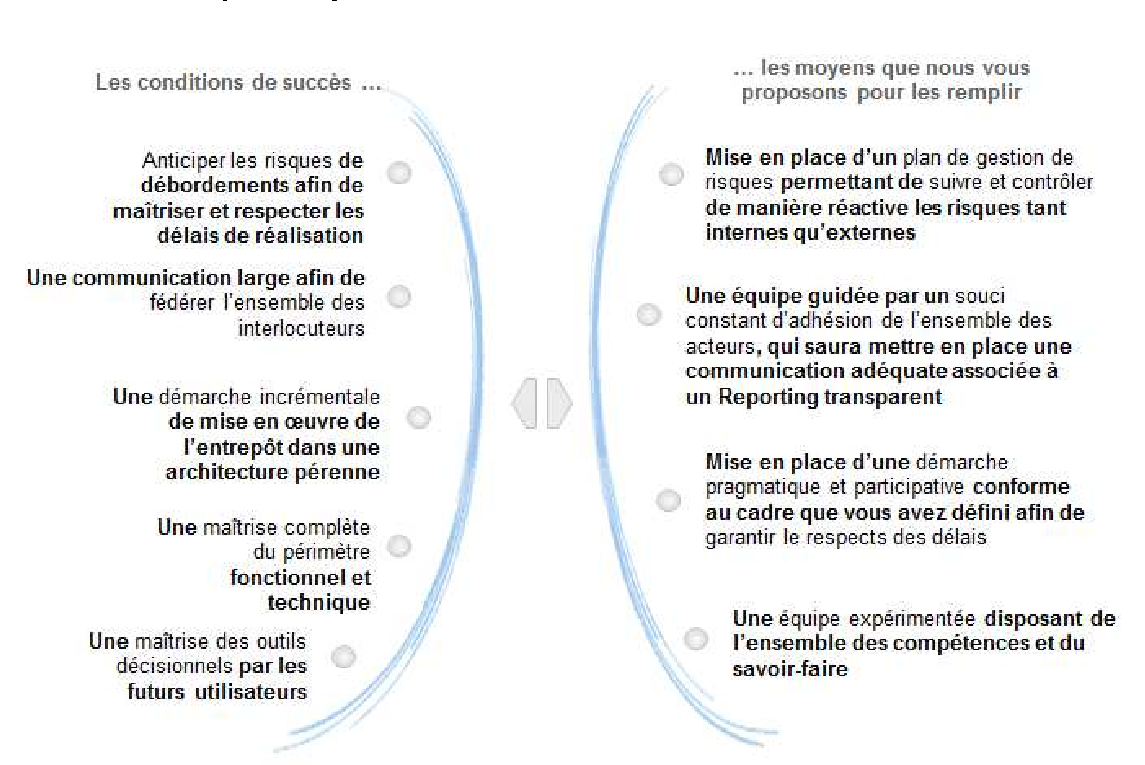
Voici un point important qu'il ne faut pas négliger. Voici donc quelques extraits des clés du succès. Faites les adaptations en fonction de votre contexte.

Facteurs clés de réussite

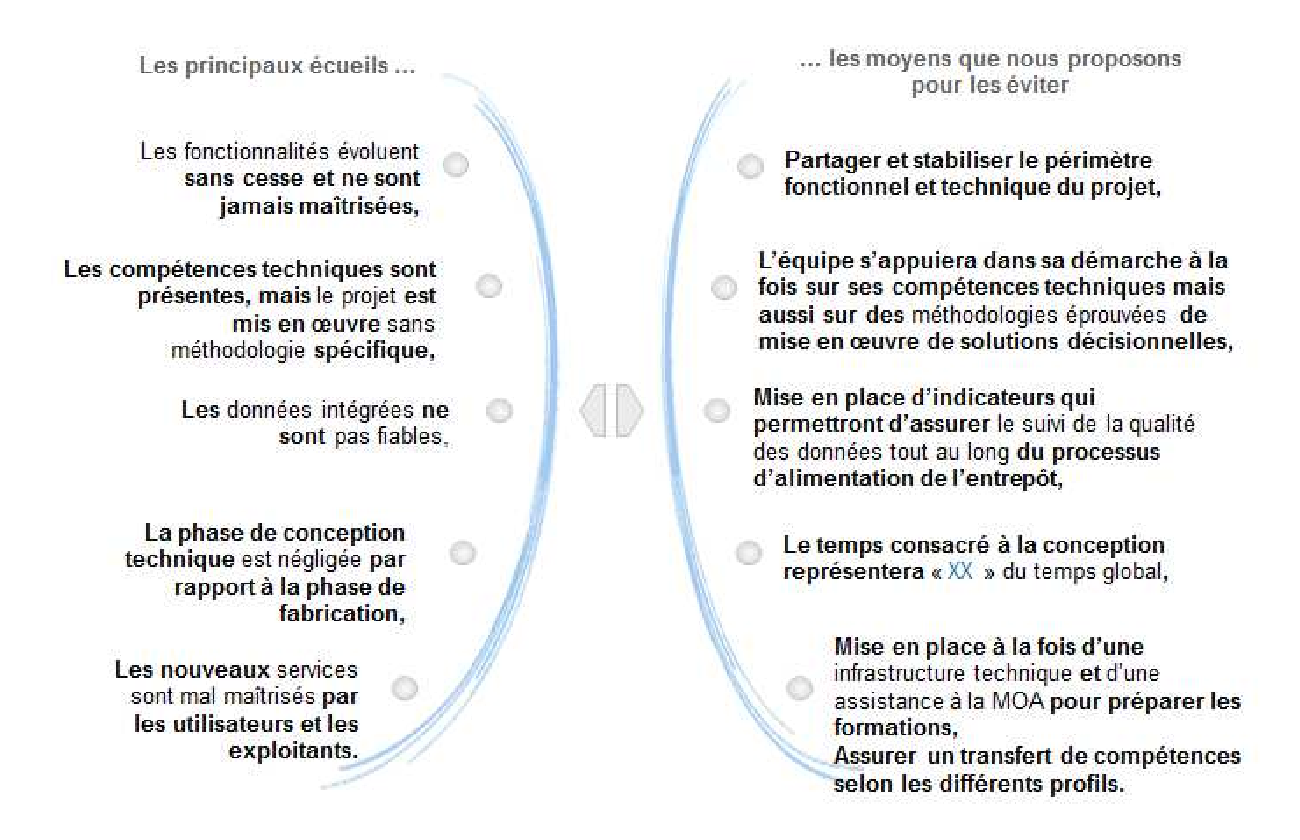
Afin de couvrir au mieux les risques inhérents à la mise en œuvre du SID au sein de « Société », notre expérience nous amène à identifier les facteurs clés de succès suivants :



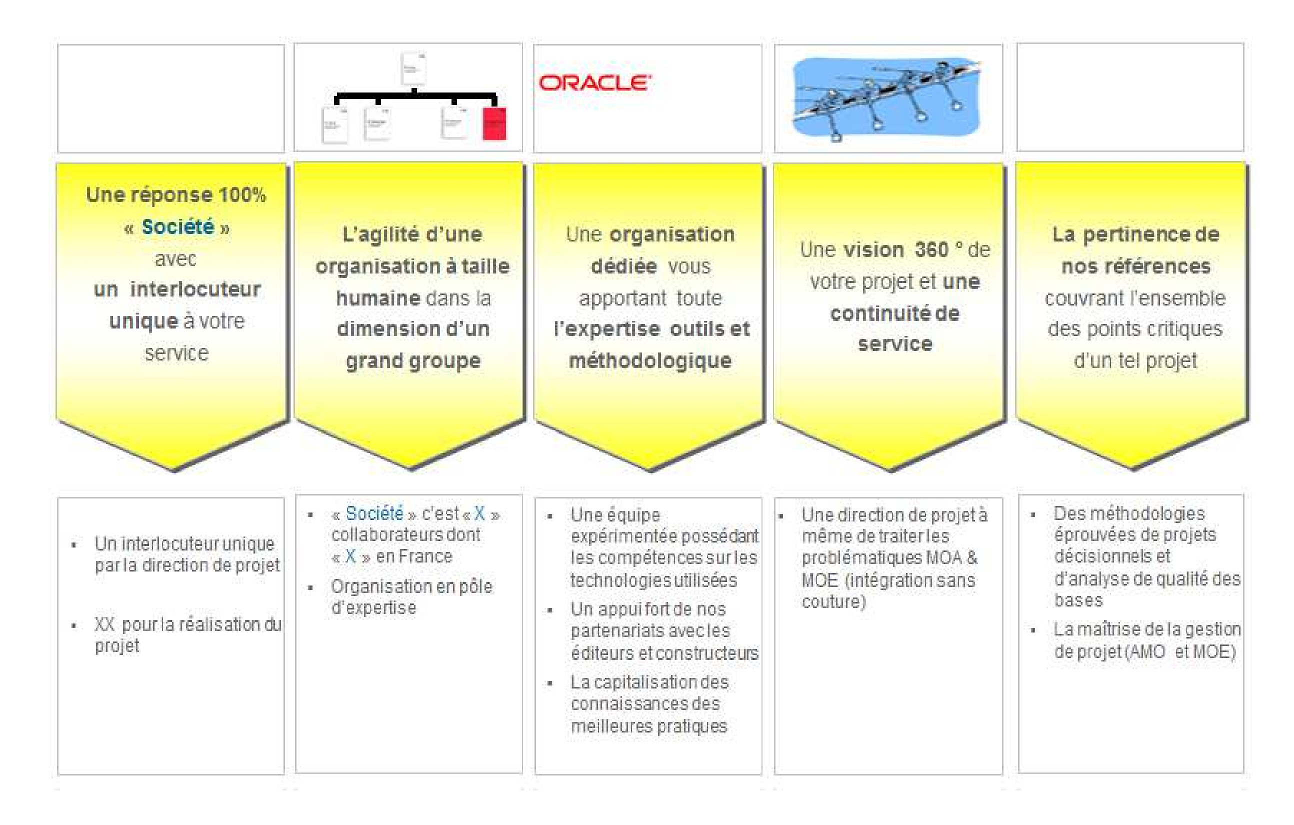
S'assurer des prés requis



Les principales causes d'échec des projets décisionnels



Les 5 atouts de XX pour vous accompagner



FACTEURS CLES EN MODE LISTE

Les facteurs clés de réussite de ce lot sont :

• La disponibilité des différents interlocuteurs fonctionnels et techniques,
• L'accès aux systèmes documentaires lié au projet,
• La maîtrise du périmètre fonctionnel par la sensibilisation des utilisateurs aux enjeux et contraintes,
• La prise en compte des projets et SI connexes, par l'identification et l'analyse d'impact des projets connexes sur le SID
• La coordination des démarches métiers et SI, par la mise en place d'une démarche de convergence avec le plan d'urbanisme du SI et l'intégration des contraintes techniques (SSO, flux…)

RECUEIL DE BESOINS

Le recueil de besoins fait partie des chapitres quasiment obligatoires dans une réponse à appel d'offres ou dans un cahier des charges.

La phase consiste à qualifier les besoins fonctionnels et à analyser l'existant afin de fournir le cahier des charges des spécifications détaillées techniques. Elle contient principalement les spécifications fonctionnelles détaillées et les spécifications techniques générales en adéquation avec la cible du client.

OBJECTIFS

Les objectifs sont de garantir la couverture fonctionnelle de l'ensemble des processus métiers concernés dans le périmètre défini. Il importe donc d'assurer un balayage global des besoins décisionnels dans toute leur transversalité.

Pour répondre à ces objectifs, différents volets d'analyse seront menés dans le cadre de ce lot, à savoir la définition des besoins fonctionnels, l'analyse de l'existant et la conception de la cible globale, pour avoir le dossier fonctionnel des besoins (fonctionnels et techniques).

Les 2 volets importants de la spécification des besoins des utilisateurs sont :

Recensement des besoins fonctionnels

• Recensement des méthodes de recueil des besoins et des supports
• Recensement, Identification et Évaluation des besoins fonctionnels
• Rédaction des documents de recueil et documents de synthèse :

- Périmètres fonctionnels
- Niveau territorial
- Indicateurs
- Acteurs du SID

Analyse de la couverture fonctionnelle

• Comparaison des besoins d'informations existantes et futures
• Analyse de l'existant : À qualifier et analyser les données du SID (bases de données existantes ou en cours de construction)
• Définition des actions à mener pour répondre aux besoins exprimés.

METHODOLOGIE

Les phases de recensement des besoins fonctionnels et d'analyse de la couverture fonctionnelle existante et future seront menées conjointement. À l'issue de ces phases, la solution cible sera identifiée à travers la définition des besoins fonctionnels. Une part du recensement se fait avec la population des utilisateurs à travers les entretiens et ateliers.

Population à rencontrer

La population concernée à minima par le recensement et l'analyse des besoins de spécifications est représentée par 3 directions :

• Direction générale
• Direction du contrôle des finances
• Direction des ressources humaines

Macro-mode opératoire

Le tableau suivant présente le croisement entre les lignes organisationnelles à rencontrer et le mode de recueil de besoins pour chaque direction.

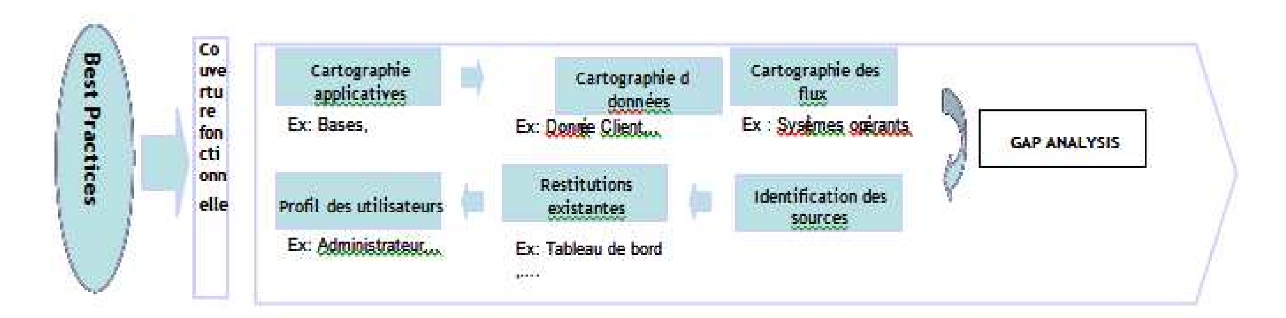
Direction / Organisation	Direction Générale	Direction du contrôle des finances	Direction des ressources humaines
Entretiens	1	1	1
Ateliers	1	1	1

En fonction de la charge de travail, des ateliers ou entretiens complémentaires peuvent être définis communément au début de la mission.

Analyse de la couverture fonctionnelle

La phase d'analyse de la couverture fonctionnelle permet de dresser un état des lieux du SI en termes d'existants et de cible par rapport aux besoins exprimés. Elle prendra la forme d'entretiens individuels avec les experts techniques et fonctionnels responsables des différents systèmes et d'analyse documentaire. Il sera utile de prévoir une mise à disposition de la documentation des différents systèmes, ainsi qu'un éventuel accès (restreint/encadré) aux systèmes eux-mêmes. L'analyse de l'existant possède les objectifs suivants :

• Formalisation de la couverture fonctionnelle de l'existant (Processus existant, cartographie applicative, cartographie des données, cartographie des flux, outils de restitutions existant…)
• Formalisation des limites et axes d'amélioration



La phase d'analyse de l'existant présentera le SI existant en termes d'applications décisionnelles et apportera les axes d'amélioration éventuels. Cette phase permettra de définir le cadre de réalisation de la solution cible en tenant compte des projets connexes.

Recensement des besoins fonctionnels

Constitution du binôme : Avant toutes choses, il est important de mentionner la constitution d'un binôme affecté à chaque exercice de recensement des besoins par un entretien ou bien un atelier thématique. Ce binôme sera choisi à partir de l'équipe projet, et sera déterminé en fonction :

• Du thème métier abordé en atelier
• Des personnes à rencontrer
• Du niveau hiérarchique des personnes présentes dans les ateliers

L'adéquation des profils de l'équipe XXXX avec les interlocuteurs de l'YYYY sera préalablement validée par la Direction de Projet.

Nous proposons les hypothèses structurales suivantes pour la réalisation des spécifications de besoins. Nous procéderons par entretien ou par atelier en fonction des cas de figure. Les paragraphes suivants présentent ces deux modalités de réalisation.

Modalités relatives aux entretiens : Les modalités des entretiens seront abordées au regard de la population concernée, de la démarche préconisée et des grilles d'entretiens utilisées.

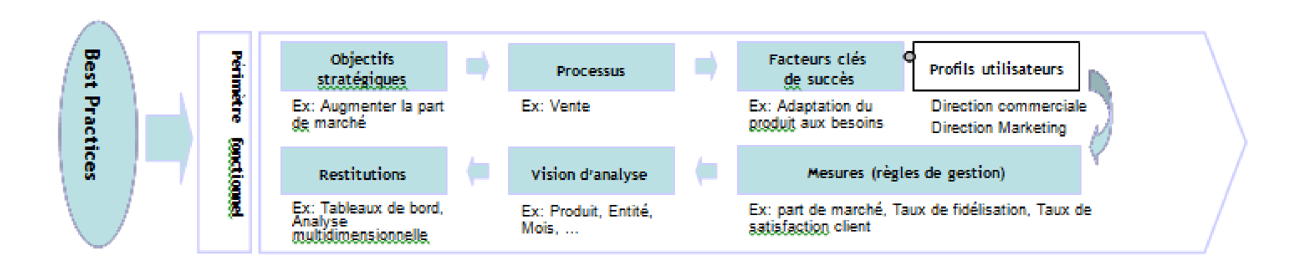
Population concernée : Les utilisateurs pouvant potentiellement être sollicités pour des entretiens sont des interlocuteurs « privilégiés » dans le projet :

• De par son niveau hiérarchique,
• De par son statut d'utilisateur clé (représentatif),
• De par son niveau d'expertise (fonctionnel ou technique).

Démarche des entretiens : Les entretiens peuvent être individuels avec au maximum 2 personnes. Ces entretiens utilisés permettront :

• De présenter la méthode utilisée, précision des « règles du jeu »
• De faire expliciter par les utilisateurs leurs « quotidiens », leurs besoins, leurs exigences et attentes
• D'identifier les relations de travail au sein de l'organisation et les difficultés rencontrées
• De valider les informations structurées recueillies

L'objectif principal est d'aider l'interviewé à formuler et exprimer son besoin de pilotage. Tout entretien fera l'objet d'une phase de préparation, pour définir le canevas précis pour formaliser le besoin à savoir la définition des thèmes que l'on va aborder en fonction de l'objectif à atteindre. Les thèmes seront déclinés en questions qui serviront de fil conducteur tout au long de l'entretien. Les thèmes abordés couvriront l'ensemble des phases de la méthodologie proposée, de la définition des objectifs stratégiques à la définition des restitutions.



Exemple de grilles d'entretien : Les objectifs de ce questionnaire sont les suivants :

• Appréhender le métier et l'organisation de la société
• Collecter les éléments permettant d'effectuer l'analyse du SID existant
• Rassembler les informations majeures pour l'élaboration du modèle du SID cible

La grille d'entretien sera la base de l'entretien. La liste suivante présente un exemple de questions types pouvant être abordées :

- Vision technique et exemple de grille d'entretien : L'objectif est ici un état des lieux du SI existant sous l'angle technique, au travers d'entretiens avec la DSI et la cellule en charge de l'urbanisation.

THEMES	CONTENU
Généralités	Nom de l'interlocuteur Position et rôle au sein de la société Relation avec les différentes maîtrises d'ouvrages et/ou d'œuvres …
Description de l'architecture métier	Domaines métiers Couverture fonctionnelle
Description de l'architecture applicative	Nom de ou des applications Technologies Nombre d'utilisateurs potentiels Volume de données traitées Flux entrants …
Description de l'architecture technique	Technologie (éditeurs, etc.)
Description des rapports (identifier le nombre de rapports par application)	Nombre de rapports Fréquence de « diffusion » Niveau de complexité « de lecture » Interface utilisateur (c/s, web...)
Données – Référentiels (identifier les sources de données pour les rapports listés ci-dessus)	Technologie Type de stockage (fichier plat, relationnel, multidimensionnel) Volume traité Performance ? (satisfaisante, insatisfaisante) Niveau de qualité actuel (satisfaisant, insatisfaisant)
Quels sont les projets en cours ?	Question ouverte
Selon vous quels sont les risques sur un tel projet ?	Question ouverte
Prés requis fonctionnels	De quel type d'information les personnes des domaines métier cibles ont-elles besoin ? Quels rapports veulent-elles ? Quels rapports sont les plus importants et quels sont ceux les moins importants ? Quel type de requêtes les utilisateurs pourront-ils lancer ? Qui administrera les données, les univers... ?

	Existe-t-il au sein du Système d'information des requêtes déjà similaires ?
Pré requis pour les données	Quelles données métier les utilisateurs veulent-ils ? Où trouvent-ils ces données aujourd'hui? Comment les données sont-elles « nettoyées » aujourd'hui, et comment doivent-elles être dans le futur ? Quelles données sont considérées comme les plus importantes pour les domaines métiers? Les données peuvent-elles être agrégées, si oui, quelles sont les dimensions ? Les utilisateurs doivent-ils pouvoir « descendre » dans le détail, si oui, avec quelle granularité ? Existe-t-il des utilisateurs pouvant valider les données, si oui quand sont-ils disponibles ? Quels sont les délais en termes de restitution des données ?
Pré requis en termes d'historique	Quelle est la durée de stockage souhaitée ? Existe-t-il un historique des données à disposition ?
Pré requis sécurité	Quelle est la sécurité qui doit être appliquée ? Quelle typologie de sécurité existe-t-il aujourd'hui ? La sécurité doit-elle être la même pour toute la population cible ? Qui aura accès aux données ?
Pré requis en termes de performance	Quel est le délai maximum que les utilisateurs sont-ils prêts à accepter ? Quels sont les rapports qui peuvent être lancés de nuit ? Combien de fois dans la journée les utilisateurs sont-ils susceptibles de lancer un rapport ?
Données sources	Sait-on où sont stockées les données sources ? Est-il possible d'avoir plusieurs sources pour une même donnée ? Les données font-elles déjà partie intégrante d'une base de données dans un modèle relationnel, etc. ? Connaît-on les utilisateurs/propriétaires des données sources ? Existe-t-il une norme d'échange d'information au sein de la société ?
Qualité des données :	Connaît-on le niveau de qualité des données sources ? Quel est le niveau de qualité attendu en rapport avec les besoins métier ? Qui « détient » les règles métier pour valider les données ?
Nettoyage des données	Existe-t-il de la documentation sur des opérations passées de nettoyage ou bien de gestion des rejets ? Existe-t-il des tables de références ou de transcodage d'un système à un autre ? Qui connaît les erreurs les plus courantes survenues ?

- Vision fonctionnelle et exemple de grille d'entretien : L'objectif est ici un état des lieux du SI existant sous l'angle fonctionnel, au travers d'entretiens avec les domaines métiers (direction générale, contrôle de finances, ressources humaines)

THEMES	CONTENU
Généralités	Nom de l'interlocuteur Relation avec les différentes maîtrises d'ouvrages et/ou d'œuvres
Présentation du département, service, équipe	Fonction et organisation du département, service, équipe Intégration au sein de l'organigramme de la société
Analyse de l'existant	Description succincte l'activité actuelle (fonctionnalités) Quels sont vos objectifs stratégiques ? (Augmenter mes ventes...) Comment se structure votre activité (identification du processus) ? Quelles sont les informations, indicateurs que vous suivez ? (Périodicité, fiabilité...) ?

	Avez-vous des documents (référentiels, rapports type…) décrivant ou faisant référence votre utilisation? Quels sont les différents outils (SI, applications…) et les différentes bases de données que vous utilisez ? Quelles données vous remplissez dans l'application ? Quelles données vous consultez dans l'application ? De façon opérationnelle est-ce que l'outil décisionnel correspond à la réalité aujourd'hui ? (par rapport à votre mode de fonctionnement, par rapport à son positionnement comme outil de pilotage, par rapport aux indicateurs de performance)
Pistes de réflexion pour la cible	Quels sont vos besoins en termes de restitutions (que souhaitez-vous suivre avec quels types de restitutions)? Vos besoins sont-ils aujourd'hui couverts? À quoi attribuez-vous le fait que tous vos besoins ne sont pas couverts (systèmes, type du modèle…) ? Quels sont les projets en cours qui peuvent avoir une incidence sur vos besoins de restitutions? (citez tous vos projets en cours) Avec l'objectif de suivre vos processus, quels sont les indicateurs qui vous voudront être capables de suivre ?

À la fin de l'entretien, la personne rencontrée est informée qu'elle recevra un compte rendu afin qu'il puisse valider les informations recueillies. Un même utilisateur pourra être rencontré plusieurs fois :

1. Pour l'analyse de l'existant et le recueil des besoins (cibles)
2. Pour valider et prioriser les indicateurs collectés.

Modalités relatives aux ateliers thématiques

Les ateliers thématiques constituent le second levier pour recueillir les besoins. Ces ateliers réuniront au maximum 4 personnes qui occupent une même position hiérarchique et qui ont en commun une proximité fonctionnelle (métier).

Le partage d'une même position hiérarchique sera un facteur de succès dans la facilité d'expression du besoin, alors que la proximité métier sera le socle commun à partir duquel les utilisateurs dialogueront et échangeront leur vision du besoin et de la cible.

Règles essentielles régissant les ateliers de travail : Ces trois règles peuvent être représentées comme suit :

• Savoir se projeter : les tableaux de bord cible sont élaborés sans prendre en compte la faisabilité technique actuelle.
• Savoir s'exprimer : toutes les idées exprimées sont susceptibles d'alimenter la réflexion sur l'élaboration du modèle décisionnel.
• Savoir partager : les ateliers de travail doivent s'appuyer sur les expériences de chaque participant (indicateurs créés, outils spécifiques développés, pratiques…) qui peuvent être mises en commun appliqué à la solution cible.

Les facteurs clés de succès des indicateurs et des tableaux de bord définis sont :

• Représentativité
• Cohérence
• Synthèse
• Facilité d'utilisation
• Anticipation

<u>Distinction des domaines métiers projet et contenu des ateliers thématiques</u> : Selon les priorités et des délais, le mode opératoire sera différent selon les domaines abordés. Deux modes peuvent être identifiés en distinguant le domaine fonctionnel « Direction Générale » des deux autres que sont le « Contrôle de finances » et les « Ressources Humaines ». Pour la Direction Générale, les domaines Contrôle de Finance et Ressources Humaines, un minimum de 1 atelier thématique par domaine est préconisé pour recueillir les besoins fonctionnels :

• Le premier atelier a pour objectif de :

> Appeler le contexte, les attentes et les objectifs de la réunion et du projet
> Démarrer la discussion par l'analyse de l'existant
> Favoriser le brainstorming et les échanges entre utilisateurs
> Emettre et argumenter l'expression des besoins
> Obtenir l'opinion collective sur les besoins exprimés

• Le second atelier a pour objectif de :

> Synthétiser les résultats des deux premiers ateliers
> Poursuivre le brainstorming
> Compléter et affiner les besoins et indicateurs nécessaires
> Valider et prioriser les résultats fonctionnels cibles

Cependant, les ateliers ne se limitent pas aux principaux domaines fonctionnels précités.

Nous préconisons également :

• Pour la DSI : 1 atelier thématique autour de l'analyse des systèmes sources, autour de l'alimentation, du stockage et de la restitution, afin de verrouiller la compréhension et l'analyse de l'existant, ainsi que de bien prendre en compte les risques et opportunités.
• Interlocuteurs du Plan d'urbanisation : 1 atelier est proposé afin d'optimiser la prise en compte à la fois des contraintes et des orientations de l'urbanisation cible.

Tous les ateliers seront suivis d'une rédaction de compte-rendu, puis d'une validation à soumettre aux participants de l'atelier.

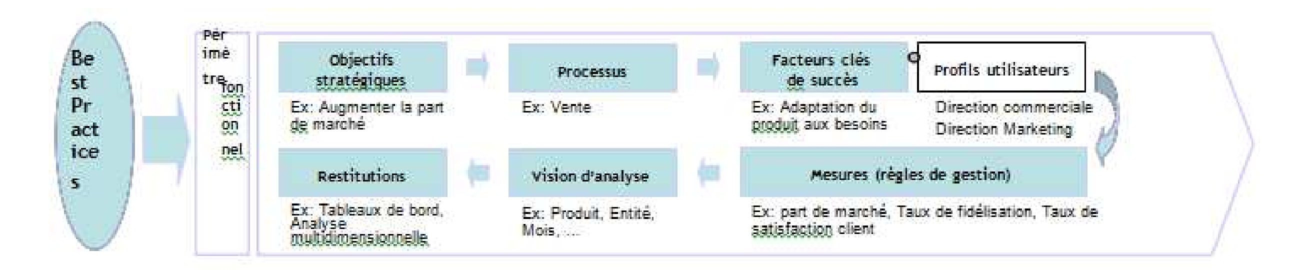
Un croisement sera ensuite effectué entre les résultats des ateliers et ceux des entretiens. Dans certains cas, les entretiens avec les experts seront la base de départ des ateliers, alors que dans d'autres les résultats des ateliers seront arbitrés par les experts désignés. De ce fait, le mode opératoire garantit l'exhaustivité, la fiabilité et la faisabilité des besoins exprimés.

Démarche du recensement des besoins

Afin de couvrir la phase de recensement des besoins et d'en faciliter le processus, nous nous appuierons sur une méthodologie d'entretien robuste : cette démarche repose à la fois sur la présence d'une grille d'entretien et sur des étapes clés à suivre pour optimiser l'expression des besoins :

1. Formaliser des objectifs stratégiques,
2. Identifier et analyser les processus métiers en relation avec ces objectifs,
3. Identifier et définir les profils utilisateurs concernés à travers le processus,

4. Choisir et définir les mesures pertinentes au regard des objectifs stratégiques en termes d'axes et d'indicateurs de pilotage,
5. Définir les regroupements indicateurs axes en termes de restitutions,
6. Définir la solution cible générale en formalisant les besoins.

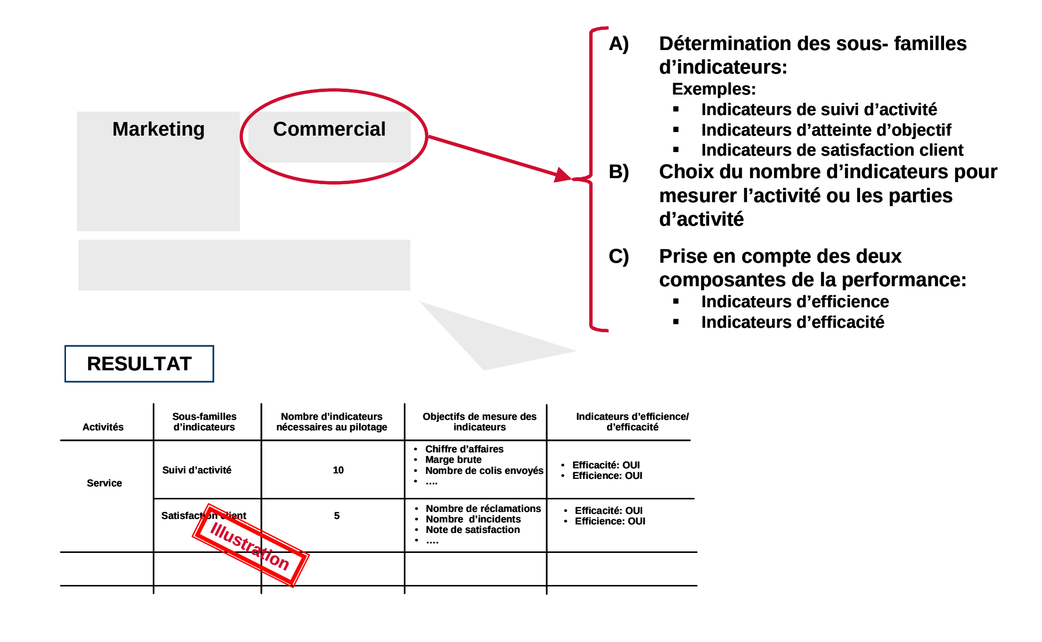


Cette méthodologie dresse la trame des entretiens qui seront menés avec les utilisateurs identifiés dans le périmètre de projet.

La base des grilles d'entretien sera adaptée à chaque cible en termes de métier, et de profil. Nous vous avons placé dans ce dossier deux grilles d'entretien à titre d'exemple dans le paragraphe « Exemple de grilles d'entretiens » ci-avant.

Illustration de la démarche du recensement des besoins

À partir d'un domaine métier comme celui de la Direction Commerciale, le cheminement méthodologique (décrit ci-dessus) débute par l'identification des objectifs stratégiques et facteurs clés de succès, aboutissant aux indicateurs qui permettent leur suivi. Ceci est matérialisé par l'exemple ci-dessous (intégrant également une illustration du fichier de travail à constituer) :

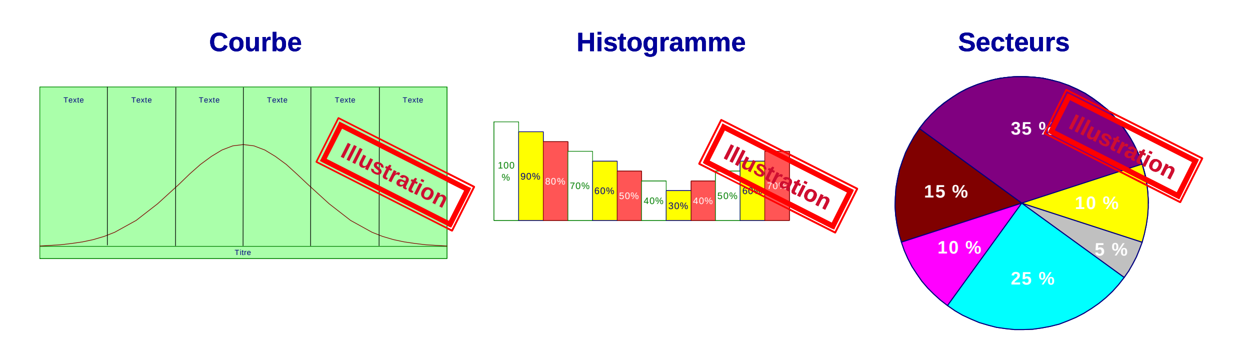


Activités	Sous-familles d'indicateurs	Nombre d'indicateurs nécessaires au pilotage	Objectifs de mesure des indicateurs	Indicateurs d'efficience/ d'efficacité
Service	Suivi d'activité	10	• Chiffre d'affaires • Marge brute • Nombre de colis envoyés •	• Efficacité: OUI • Efficience: OUI
	Satisfaction client	5	• Nombre de réclamations • Nombre d'incidents • Note de satisfaction •	• Efficacité: OUI • Efficience: OUI

À partir de l'identification des objectifs, facteurs clés de succès impliquant des indicateurs de suivis, il est maintenant important de détailler chaque indicateur exprimé. Nous prenons l'exemple de l'indicateur « Nombre d'incidents courrier » pour illustrer cette étape :

Indicateurs	Objectif de l'indicateur	Mode de calcul	Périodicité	Prévision / suivi
Nombre d'incidents courrier	Mesurer la qualité de service pour les prestations courrier	Nombre de retours client de nature « Incidents client »	Hebdomadaire	Indicateur de suivi

Ensuite, pour chaque indicateur détaillé, le tableau de bord associé doit être étudié afin de réfléchir sur la représentation de cet indicateur sous forme de :

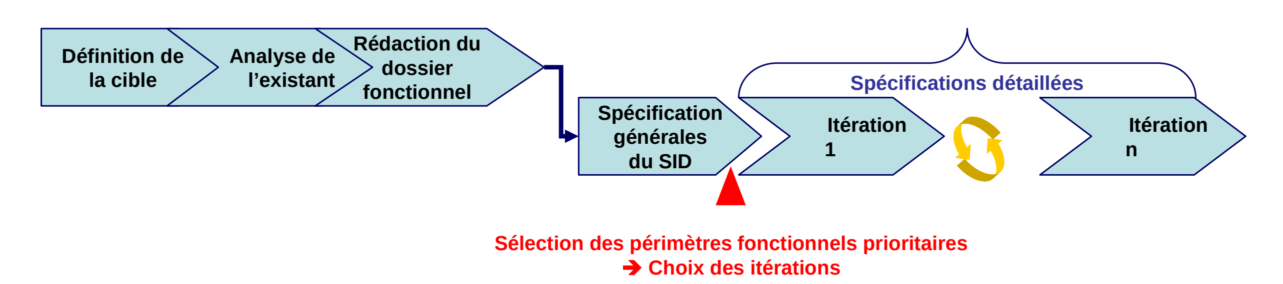


Enfin, la priorisation des indicateurs recueillis est un élément consensuel à aborder et à valider : les participants des ateliers doivent classer chacun des indicateurs choisis en fonction de leur importance pour le pilotage du domaine métier concerné (le domaine commercial dans notre illustration) :

• Haute priorité : La Direction peut difficilement se passer d'un tel indicateur pour piloter son activité
• Priorité moyenne : Cet indicateur est important pour le pilotage de la Direction
• Faible priorité : La Direction peut être pilotée sans la présence de cet indicateur

Définition de la cible générale et périmètre de la première itération

À l'issue des phases de recensement des besoins, de l'analyse de la couverture fonctionnelle, il faut réaliser l'étude de dimensionnement et identifier le périmètre fonctionnel de la première itération.



Nous préconisons une mise en œuvre par lots fonctionnels (ref : Facteurs clés de succès du projet SID) en phase avec le cahier des charges de YYYY. Aussi l'un des objectifs est de déterminer le périmètre fonctionnel de la première itération du projet SID (version de démarrage).

Ce périmètre sera défini en utilisant une analyse de la valeur (utilité et urgence versus complexité).

Ce périmètre restreint fera l'objet d'un cahier des charges fonctionnelles, premier jalon de la mise en œuvre.

Le schéma ci-dessus présente notre démarche de la définition de la cible à la sélection du périmètre fonctionnel prioritaire pour la première itération.

L'ensemble de ces éléments permettent de définir la couverture fonctionnelle cible en termes applicatifs, techniques et organisationnels.

À l'issue de cette phase, le dossier fonctionnel spécifiera les besoins fonctionnels et techniques du SID. Ce dossier décline l'atteinte de la cible dans le respect du cadre du schéma global d'urbanisation du SI.

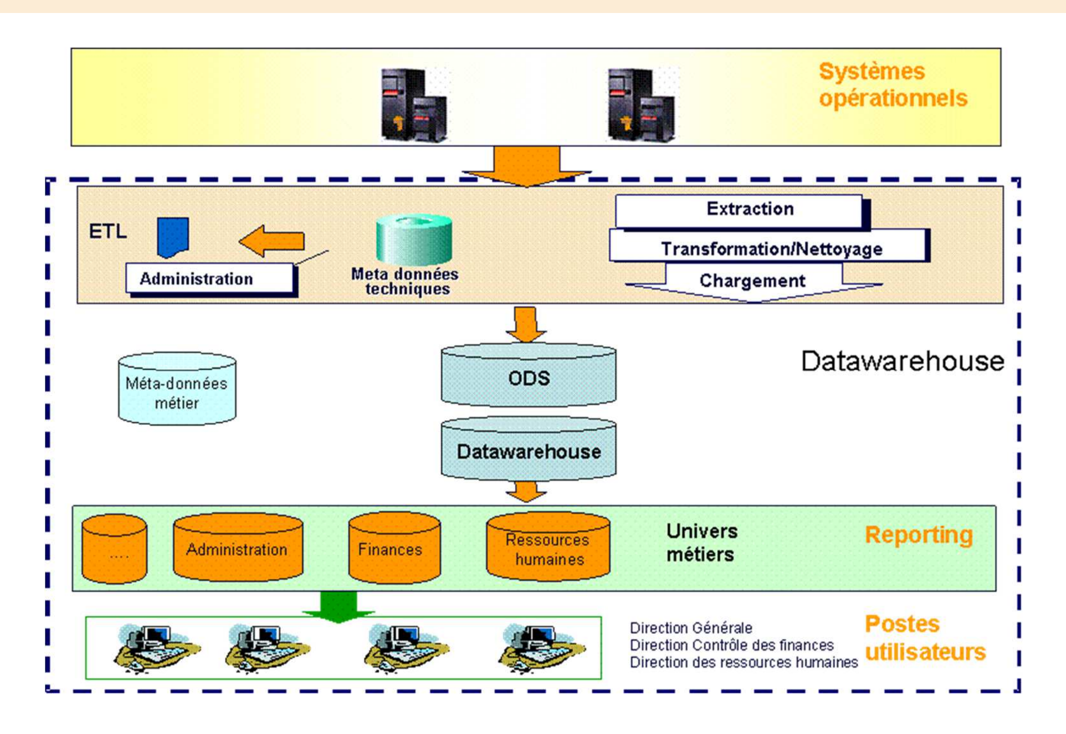
La première itération identifiée permet de définir le premier périmètre fonctionnel à prendre en compte. Il faut alors sélectionner le couple éditeur/ intégrateur le plus en adéquation avec les besoins.

En fonction du périmètre technique et fonctionnel de l'itération (multidimensionnel, datamining, requêteur…) il faudra réaliser un choix de couple éditeur/ intégrateur répondant aux besoins.

<u>Exemple</u> :

• L'itération 1 concerne l'alimentation, le stockage et la présentation via des outils multidimensionnels avec possibilités de réaliser des simulations.
• L'itération 2 concerne l'alimentation, le stockage et la présentation via des outils Datamining.

DÉMARCHE DE LA CONCEPTION CIBLE



Facteurs clés de succès

Les facteurs clés de réussite de ce lot sont :

- La disponibilité des différents interlocuteurs fonctionnels et techniques,
- L'accès aux systèmes documentaires liés au projet,
- La maîtrise du périmètre fonctionnel par la sensibilisation des utilisateurs aux enjeux et contraintes,
- La prise en compte des projets et SI connexes, par l'identification et l'analyse d'impact des projets connexes sur le SID
- La coordination des démarches métiers et SI, par la mise en place d'une démarche de convergence avec le plan d'urbanisme du SI et l'intégration des contraintes techniques (SSO, flux,…)

Livrables

Le tableau suivant récapitule les livrables en termes d'approche méthodologique.

		Approche méthodologie				
Ref.	**Livrables**	**Entretiens**	**Réunion de travail / Ateliers**	**Synthèse documentaire et analyse**	**Best Practices BT CSI**	**Méthodologie de cadrage**
0.1	Support de réunion de lancement	x	x	x	Note de lancement type	
1.1	Entretien de recensement :	x	x	x	Grilles d'entretiens Connaissance de la Poste	x
	Document d'information aux participants des entretiens	x	x	x		
	Grilles d'entretien	x	x	x		
	Ordre du jour			x		
	Compte-rendu de réunions (document de recueil,…)			x		
1.2	Document de recensement des besoins fonctionnels:	x	x	x	Plans types	x
	Spécification des axes et mesures	x	x	x		x
	Spécification des tableaux de bord	x	x	x		x
	Identification des données et règles de gestion	x	x	x		x
1.3	Dossier d'analyse de l'existant:	x	x	x	Grilles d'entretiens Méthodologie de cartographie	x
	Cartographie applicative	x	x	x		x
	Cartographie des données	x	x	x		x
	Cartographie des flux	x	x	x		x
	Restitutions existantes	x	x	x		x
1.4	Etude de dimensionnement	x	x	x	Plan type Best practices issues d'expériences de cadrage	x
1.5	Dossier fonctionnel et technique du SID:	x	x	x	Plan type Best practices issues d'expériences de cadrage	x
	Identification de l'origine et de la disponibilité des données		x	x		x
	Schéma des flux		x	x		x
	Macro- modèle		x	x		x
	Architecture cible applicative et technique		x	x		x
	Plan de réalisation		x	x		x
	Dossier fonctionnel détaillé de la première itération		x	x		x
1.6	Dossier fonctionnel détaillé de la première itération		x	x	Plan type Best practices issues d'expériences de cadrage	x

Livrables proposés en compléments du dossier de consultation

GUIDE D'ENTRETIENS

Les objectifs de ce questionnaire sont les suivants :

- Appréhender le métier et l'organisation
- Collecter les éléments permettant d'effectuer l'analyse du SID existant
- Rassembler les informations majeures pour l'élaboration du modèle du SID cibles

Objectif : Faire un état des lieux du SI existant (vision technique)

THEMES	
Généralités	Nom de l'interlocuteur Position et rôle Relation avec les différentes maîtrises d'ouvrages et/ou d'œuvres …
Description de l'architecture métier	Domaines métiers Couverture fonctionnelle
Description de l'architecture applicative	Nom de ou des applications Technologies Nombre d'utilisateurs potentiels Volume de données traitées Flux entrants...
Description de l'architecture technique	Technologie (éditeurs...)
Description des rapports (identifier le nombre de rapports par application)	Nombre de rapports Fréquence de « diffusion » Niveau de complexité « de lecture » Interface utilisateur (c/s, web, etc..)…
Données – Référentiels (identifier les sources de données pour les rapports listés ci-dessus)	Technologie Type de stockage (fichier plat, relationnel, multidimensionnel) Volume traité Performance ? (satisfaisante, insatisfaisante) Niveau de qualité actuel (satisfaisant, insatisfaisant)
Quels sont les projets en cours ?	
Selon vous quels sont les risques sur un tel projet ?	
Prés requis fonctionnels	De quel type d'information les personnes des domaines métier cibles ont-elles besoin ? Quels rapports veulent-elles ? Quels rapports sont les plus importants et quels sont ceux les moins importants ? Quel type de requêtes les utilisateurs pourront-ils lancer ? Qui administrera les données, les univers, etc.? Existe-t-il au sein du Système d'information des requêtes déjà similaires ?
Pré requis pour les données	Quelles données métier les utilisateurs veulent-ils ? Où trouvent-ils ces données aujourd'hui? Comment les données sont-elles « nettoyées » aujourd'hui, et comment doivent-elles être dans le futur ? Quelles données sont considérées comme les plus importantes

	pour les domaines métiers? Les données peuvent-elles être agrégées, si oui, quelles sont les dimensions ? Les utilisateurs doivent-ils pouvoir « descendre » dans le détail, si oui, avec quelle granularité ? Existe-t-il des utilisateurs pouvant valider les données, si oui quand sont-ils disponibles ? Quels sont les délais en termes de restitution des données ?
Pré requis en termes d'historique	Quelle est la durée de stockage souhaitée ? Existe-t-il un historique des données à disposition ?
Pré requis sécurité	Quelle est la sécurité qui doit être appliquée ? Quelle typologie de sécurité existe-t-il aujourd'hui ? La sécurité doit-elle être la même pour toute la population cible ? Qui aura accès aux données ?
Pré requis en termes de performance	Quel est le délai maximum que les utilisateurs sont-ils prêts à accepter ? Quels sont les rapports qui peuvent être lancés de nuit ? Combien de fois dans la journée les utilisateurs sont-ils susceptibles de lancer un rapport ?
Données sources	Sait-on où sont stockées les données sources ? Est-il possible d'avoir plusieurs sources pour une même donnée ? Les données font-elles déjà partie intégrante d'une base de données dans un modèle relationnel, etc. ? Connaît-on les utilisateurs/propriétaires des données sources ? Existe-t-il une norme d'échange d'information au sein de la Poste ?
Qualité des données :	Connaît-on le niveau de qualité des données sources ? Quel est le niveau de qualité attendu en rapport avec les besoins métier ? Qui « détient » les règles métier pour valider les données ?
Nettoyage des données	Existe-t-il de la documentation sur des opérations passées de nettoyage ou bien de gestion des rejets ? Existe-t-il des tables de références ou de transcodage d'un système à un autre ? Qui connaît les erreurs les plus courantes survenues ?

Objectif : Faire un état des lieux du SI existant (vision fonctionnelle)

En conséquence, ce questionnaire est organisé en plusieurs parties :

· Une partie dédiée à la présentation de l'interlocuteur et de son département, service, équipe
· Une partie consacrée au modèle existant du SID
· Une partie dédiée à l'identification des axes d'améliorations alimentant la définition de la cible

THEMES	
Généralités	Nom de l'interlocuteur Relation avec les différentes maîtrises d'ouvrages et/ou d'œuvres
Présentation du département, service, équipe	Fonction et organisation du département, service, équipe Intégration au sein de l'organigramme du GMSIH
Analyse de l'existant	Description succincte l'activité actuelle (fonctionnalités) Quels sont vos objectifs stratégiques ? (Augmenter mes ventes…..) Comment se structure votre activité (identification du processus) ? Quelles sont les informations, indicateurs que vous suivez ? (Périodicité, fiabilité…) ? Avez-vous des documents (référentiels, rapports type...)

	décrivant ou faisant référence votre utilisation? Quels sont les différents outils (SI, applications…) et les différentes bases de données que vous utilisez ? Quelles données vous remplissez dans l'application ? Quelles données vous consultez dans l'application ? De façon opérationnelle est-ce que l'outil décisionnel correspond à la réalité aujourd'hui ? (par rapport à votre mode de fonctionnement, par rapport à son positionnement comme outil de pilotage, par rapport aux indicateurs de performance)
Pistes de réflexion pour la cible	Quels sont vos besoins en termes de restitutions (que souhaitez-vous suivre avec quels types de restitutions)? Vos besoins sont-ils aujourd'hui couverts? À quoi attribuer le fait que tous vos besoins ne sont pas couverts (systèmes, type du modèle…) ? Quels sont les projets en cours qui peuvent avoir une incidence sur vos besoins de restitutions? (citez tous vos projets en cours) Avec l'objectif de suivre vos processus, quels sont les indicateurs qui vous voudront être capables de suivre ?

CMMI (CAPABILITY MATURITY MODEL INTEGRATION)

CMMI, sigle de Capability Maturity Model + Integration, est un modèle de référence, un ensemble structuré de bonnes pratiques, destiné à appréhender, évaluer et améliorer les activités des entreprises d'ingénierie.

CMMI a été développé par le Software Engineering Institute de l'université Carnegie Mellon, initialement pour appréhender et mesurer la qualité des services rendus par les fournisseurs de logiciels informatiques du Département de la Défense US (DoD). Il est maintenant largement employé par les entreprises d'ingénierie informatique, les Directeurs des systèmes informatiques et les industriels pour évaluer et améliorer leurs propres développements de produits.

LE MODELE CMMI

Le modèle **CMMI** définit une échelle de mesure de la maturité à 5 niveaux, ainsi que les indicateurs nécessaires pour évaluer les activités menées par une équipe par rapport à cette échelle - l'équipe peut être un groupe de travail, un ou plusieurs projets, une société voire une institution d'État.

CMMI est un cadre générique de processus qui se décline en 3 modèles (appelés constellations) :

- **CMMI-DEV** pour le développement de systèmes (logiciel ou autre, modèle publié en août 2006)
- **CMMI-ACQ** pour la maîtrise des activités d'achat (modèle publié en novembre 2007)
- **CMMI-SVC** pour la fourniture de services (modèle publié en février 2009)

La particularité de ces 3 modèles de processus est qu'ils ont une partie commune (le noyau ou "core" en anglais) qui représente environ 60% des pratiques. D'un modèle à l'autre, les différences portent essentiellement sur la catégorie « Ingénierie » dont les pratiques varient selon l'activité concernée.

Le modèle **CMMI** est majoritairement utilisé dans des sociétés d'informatique, toutefois les principes de **CMMI** s'appliquent à n'importe quelle activité d'ingénierie : Architecture, mécanique, électronique...

MATURITE

D'après la définition donnée dans le **CMMI**, la maturité d'une organisation est le degré auquel celle-ci a déployé explicitement et de façon cohérente des processus qui sont documentés, gérés, mesurés, contrôlés et continuellement améliorés.

Un niveau de maturité (Maturity Level) correspond à l'atteinte d'un niveau de capabilité uniforme pour un groupe de processus. Un niveau de capabilité (Capability Level) mesure l'atteinte des objectifs d'un processus pour le niveau donné.

HISTORIQUE

Dans les années 1980, le Département de la Défense des États-Unis (DoD) a demandé l'élaboration d'un référentiel de critères lui permettant d'évaluer ses fournisseurs de logiciels. Après une lente maturation, le SEI (Software Engineering Institute) financé par le DoD a présenté en 1991 le Capability Maturity Model (**CMM**). Ce modèle de référence ne concerne que les bonnes pratiques du génie logiciel. Après un fort engouement pour ce modèle, d'autres modèles similaires ont vu le jour, tels que :

• **SE-CMM** (pour System Engineering) ;
• **SA-CMM** (pour Software Acquisition) ;
• **IPD-CMM** (pour Integrated Product Development) ;
• **People CMM** pour le management des ressources humaines ;
• **SS-CMM** pour Supplier Sourcing.

Tant et si bien qu'il fallut rebaptiser le **CMM** « initial » en **SW-CMM** (pour Software). En 2001, le SEI a proposé une nouvelle version de son modèle, le **CMMI** (Capability Maturity Model Integration) qui englobe les bonnes pratiques des autres modèles, sauf la gestion des ressources humaines qui n'est pas encore considérée (version 1.1). La version actuelle du modèle a été réactualisée en 2006 (version 1.2). Cette dernière version du CMMI tend à simplifier le modèle et améliore la prise en compte des composants de type matériel.

Le **CMMI-DEV** est un modèle de processus (référentiel de bonnes pratiques) pour la réalisation de tout type de produit (ou système). C'est cependant dans le développement et la maintenance de logiciel qu'il est le plus utilisé. Sa portée utile s'étendant de l'apparition d'un besoin jusqu'à la livraison du produit correspondant, il y a d'autres modèles utiles pour d'autres domaines du logiciel, par exemple pour les infrastructures et opérations ITIL.

DESCRIPTIF DU MODELE

Dans l'approche étagée (il existe une approche dite "continue"), les bonnes pratiques préconisées par le modèle (version 1.2) sont rassemblées en 22 domaines de processus eux-mêmes regroupés en 5 niveaux de maturité. Les domaines de processus rattachés à un niveau de maturité M ne peuvent être stabilisés et efficaces que si les domaines de processus des niveaux inférieurs (< M) sont déjà stabilisés et efficaces (principe d'empilement). Les 5 niveaux sont :

• **Initial (niveau de maturité 1) :** Il n'y a pas de grand pilier directionnel, aucune façon de faire ou standard ne sont établis (ou bien ils sont documentés, mais ne sont pas utilisés), tout doit être fait. Il n'y a pas de surveillance (monitoring), aucune évaluation de performance et la communication est absente. Les faiblesses ne sont pas identifiées et les employés ne sont pas au courant de leurs responsabilités de façon définie et absolue. Les réactions aux incidents se font en mode urgence, sans identification claire des priorités. À ce niveau les solutions ainsi que les projets sont décidés, développés et instaurés par un individu. Les compétences et les ressources propres de cet individu sont la raison du succès ou de l'échec du projet (par dérision, ce niveau est aussi nommé héroïque ou chaotique). Il n'y a pas de description du niveau de maturité 1 dans le modèle.

• « **managed** », soit discipliné en français (niveau de maturité 2) : Une discipline est établie pour chaque projet et se matérialise essentiellement par des plans de projet (plan de

développement, d'assurance qualité, de gestion de configuration ...). Le chef de projet a une forte responsabilité dans le niveau 2 : Il doit définir, documenter, appliquer et maintenir à jour ses plans. D'un projet à l'autre, il capitalise et améliore ses pratiques de gestion de projet et d'ingénierie.

• « **Defined** », sois ajusté en français (niveau de maturité 3) : ce niveau est caractérisé par une standardisation adéquate des pratiques, une capitalisation centralisée (en particulier sur les mesures réalisées dans les projets) et une maîtrise du référentiel interne (ou Système Qualité). Il existe des lignes directrices, un plan stratégique et une planification de l'amélioration de processus pour le futur, en ligne avec les objectifs d'affaire de l'organisation. Les employés sont formés et conscients de leurs responsabilités ainsi que de leurs devoirs.

• « **Quantitatively managed** », soit géré quantitativement en français (niveau de maturité 4) : les projets sont pilotés sur la base d'objectifs quantitatifs de qualité produit et processus. La capacité des activités (ou sous-processus) critiques est déterminée par l'organisation, ainsi que les modèles de performance et de prévision associés. L'expression de la qualité demandée par le client est prise en compte pour quantifier les objectifs du projet et établir des plans selon la capacité des processus de l'organisation.

• « **Optimizing** », soit en optimisation en français (niveau de maturité 5) : Les processus qui sont gérés quantitativement pour le pilotage de projet (niveau de maturité 4) sont en optimisation constante afin d'anticiper les évolutions prévues (besoins clients, nouvelles technologies...).

LE NIVEAU 1

Le niveau 1 initial est le niveau où le résultat final est imprévisible. À ce niveau l'effort individuel prévaut à l'effort collectif dirigé vers un but établi. L'atteinte des résultats repose plus sur les hommes, sur leur engagement et bonne volonté, que sur l'application disciplinée de bonnes pratiques. La réussite d'un projet repose en général sur le talent d'un individu, c'est pourquoi on surnomme ironiquement ce niveau l'ère des héros. Mais une réussite éventuelle ne sera pas nécessairement reproductible. L'évaluation de l'efficacité et des performances est absente. La direction n'établit pas de plan ou de vision qui sont liés à des besoins. La documentation est inexistante.

Exemple : La supervision des sauvegardes est réalisée par un technicien qui s'intéresse aux systèmes de sauvegarde. Il comprend et maîtrise ce qu'il réalise. En général il vérifie quotidiennement les résultats des traitements de la veille, personne ne le contrôle ou ne le supervise. Si nécessaire il applique les correctifs au système, néanmoins, il travaille de manière autarcique. Si pour une raison quelconque (surcharge, congés) il n'est pas en mesure de réaliser les contrôles nécessaires, ceux-ci ne sont alors pas pris en charge.

LE NIVEAU 2 (MANAGED / DISCIPLINE)

Pour le niveau 2, les activités et produits (techniques et de gestion, intermédiaire et finaux) sont maîtrisés par le projet. Les processus projet sont disciplinés, ce qui se caractérise par :

• Les activités sont planifiées et exécutées conformément à une politique (ou directive) d'organisation,
• Les rôles, responsabilités et acteurs sont définis et connus,

• Les acteurs disposent des compétences et des ressources adéquates pour réaliser les produits,
• Les produits sont contrôlés,
• La mise en œuvre du processus fait l'objet d'un suivi, de vérifications et d'ajustement si nécessaire.

Le niveau 2 se compose de sept domaines de processus traitant de :

• La gestion des exigences,
• La planification de projet,
• Le suivi de projet,
• La gestion des fournisseurs,
• L'utilisation des métriques,
• L'assurance qualité
• Et la gestion de configuration.

Chacun de ces sept domaines de processus contribue à donner à l'organisation une bonne visibilité sur ses développements : Visibilité sur le contenu, les coûts, les délais, la qualité des produits développés et des processus utilisés. Typiquement, les membres d'une équipe de développement, comme le management, connaissent l'état d'avancement de leur projet et des évolutions en cours, ainsi que ce qu'il reste à faire.

LE NIVEAU 3 (DEFINED / AJUSTE)

Le niveau 3 est ajusté. À ce niveau, l'organisation dispose d'un ensemble de processus standards qui sont ajustés par chaque projet, selon le contexte client propre, avec des règles fixées par l'organisation. Les processus standards sont développés, maintenus, supportés et leur application contrôlée par un groupe processus (le **SEPG** - Software/System Engineering Process Group ou EPG). Chaque projet capitalise son expérience et permet de bonifier le capital collectif. C'est aussi à ce niveau que les cycles de vie et processus d'ingénierie sont standardisés par typologie de projet.

Le niveau 3 met en place des boucles d'amélioration : l'expérience, les forces et difficultés rencontrées lors des développements, est capitalisée pour améliorer les développements futurs. Par exemple, des problèmes rencontrés sur un projet seront analysés dans un bilan de fin de projet pour éventuellement enrichir une base de risques types et anticiper ce problème lors de développement similaire.

Le niveau 3 repose sur les bases du niveau 2. Ainsi le domaine de niveau 2 "Gestion des exigences" est complété au niveau 3 par le domaine "Développement des exigences". Une fois que, au niveau 2, l'organisation a appris à gérer ses exigences, elle peut au niveau 3 mettre en place des pratiques d'ingénierie pour définir ces exigences. D'une manière générale, les domaines de processus de niveau 2 sont des prérequis pour les domaines de niveau 3.

Le niveau 3 se compose de quatorze domaines de processus traitant de :

• Expression des besoins (Requirements Development)
• Solution technique (Technical Solution)
• Intégration du produit (Product Integration)
• Recette technique (Vérification)
• Recette fonctionnelle (Validation)

• Gérer l’organisation des processus (Organizational Process Focus)
• Définition de l’organisation (Organizational Process Definition)
• Formation à l’organisation (Organizational Training)
• Gestion multidisciplinaire de projet (Integrated Project Management for IPPD)
• Gestion des risques (Risk Management)
• Méthode de prise de décision (Decision Analysis and Resolution)
• Organisation de l’intégration (Organizational Environment for Integration)

On trouve ainsi dans ce niveau une accumulation de choses que l’on peut regrouper sous les thèmes suivants :

• L’ingénierie des exigences, de l’expression des besoins aux spécifications et à la recette ;
• Le perfectionnement de la gestion de projet ;
• La « personnalisation » de la gestion de chaque projet et la capitalisation de l’expérience d’un projet à l’autre ;
• Les méthodes de prise de décision.

LE NIVEAU 4 (QUANTITATIVELY MANAGED / GERE QUANTITATIVEMENT)

Le niveau 4 est géré quantitativement. À ce niveau, les processus clés sont sous contrôle statistique (surveillance d’indicateurs quantitatifs, et actions correctrices si dérives).

Élimination des causes spéciales de variation.

• Ex. la capitalisation a permis d’établir la productivité moyenne du processus (taille produite/charge consommée).

Le projet se fixe un objectif de productivité en relation et prend des mesures dès qu’il y a des dérives. Les méthodes statistiques de contrôle des processus mises en place au niveau 4 permettent de focaliser les actions d'amélioration sur les pratiques pour lesquelles ces actions sont les plus utiles. Au niveau des projets, elles permettent d'identifier des activités n'ayant pas atteint des résultats attendus et ainsi prendre des actions. Par exemple, en fonction de la taille, technologie, complexité d'un nouveau composant, les tests doivent permettre d'identifier un nombre de défauts compris dans une fourchette définie. Si le nombre de défauts identifiés est en dehors de cette fourchette, une analyse sera lancée pour en comprendre les raisons. Cette fourchette est définie à partir de calculs statistiques basés sur des résultats des tests des précédents composants développés.

Le niveau 4 se compose de deux domaines de processus traitant de :

• Performance des processus (Organizational Process Performance)
• Gestion quantitative du projet (Quantitative Project Management)

LE NIVEAU 5 (OPTIMIZING / EN OPTIMISATION)

Le niveau 5 est en Optimisation. L’organisation est dans une boucle permanente d’optimisation (réduction des causes communes de variation) des processus et des technologies, optimisation basée sur des analyses coût/bénéfice. Des analyses causales statistiques menées régulièrement permettent ces améliorations. Les boucles permanentes d'optimisation sont mises en place dès le niveau 3. Au niveau 5, celles-ci se basent sur des

méthodes statistiques pour focaliser les analyses causales sur les événements dont l'analyse apportera des informations permettant d'optimiser les processus, en évitant de perdre du temps sur l'analyse d'événement apportant moins d'information.

Le niveau 5 se compose de deux domaines de processus traitant de :

• Innovation organisationnelle (Organizational Innovation and Deployment)
• Analyse causale et solution des problèmes (Causal Analysis and Resolution)

LES AUTRES COMPOSANTS

• **Les objectifs génériques :** Le modèle **CMMI** fournit 5 objectifs génériques (GG). Ces objectifs génériques (et les pratiques associées) s'appliquent à tous les domaines de processus (PA).
• **Les pratiques génériques :** Les pratiques génériques appartiennent aux objectifs génériques. Elles doivent être systématiquement implémentées pour prétendre atteindre un niveau de maturité ou de capacité.
• **Les objectifs spécifiques :** Les objectifs spécifiques sont liés à un domaine de processus (PA).
• **Les pratiques spécifiques :** Elles sont liées à un objectif spécifique, donc à un domaine de processus (PA).
• **Les produits d'activité :** Ce sont tous les éléments générés par un Projet (Plan de projet, spécification, cahier de test unitaire, revue par les pairs...)

Les autres déclinaisons

• **CMM-TSP** (Team Software Process) qui détermine les pratiques normées d'une équipe projet.
• **CMM-PSP** (Personal Software Process) qui détermine les pratiques normées d'une ressource individuelle de développement.

Pour aller plus loin

Et **l'ISO 9001** dans tout ça ?

• Les deux normes concernant les processus de développement logiciel, un article de comparaison peut être consulté Comparaison entre **ISO 9001** et **CMMI**.

CMMI et **ISO 15504** alias **SPICE**

• **CMMI**, utilisé en mode continu, est un des modèles de processus accepté par la norme **ISO 15504**.

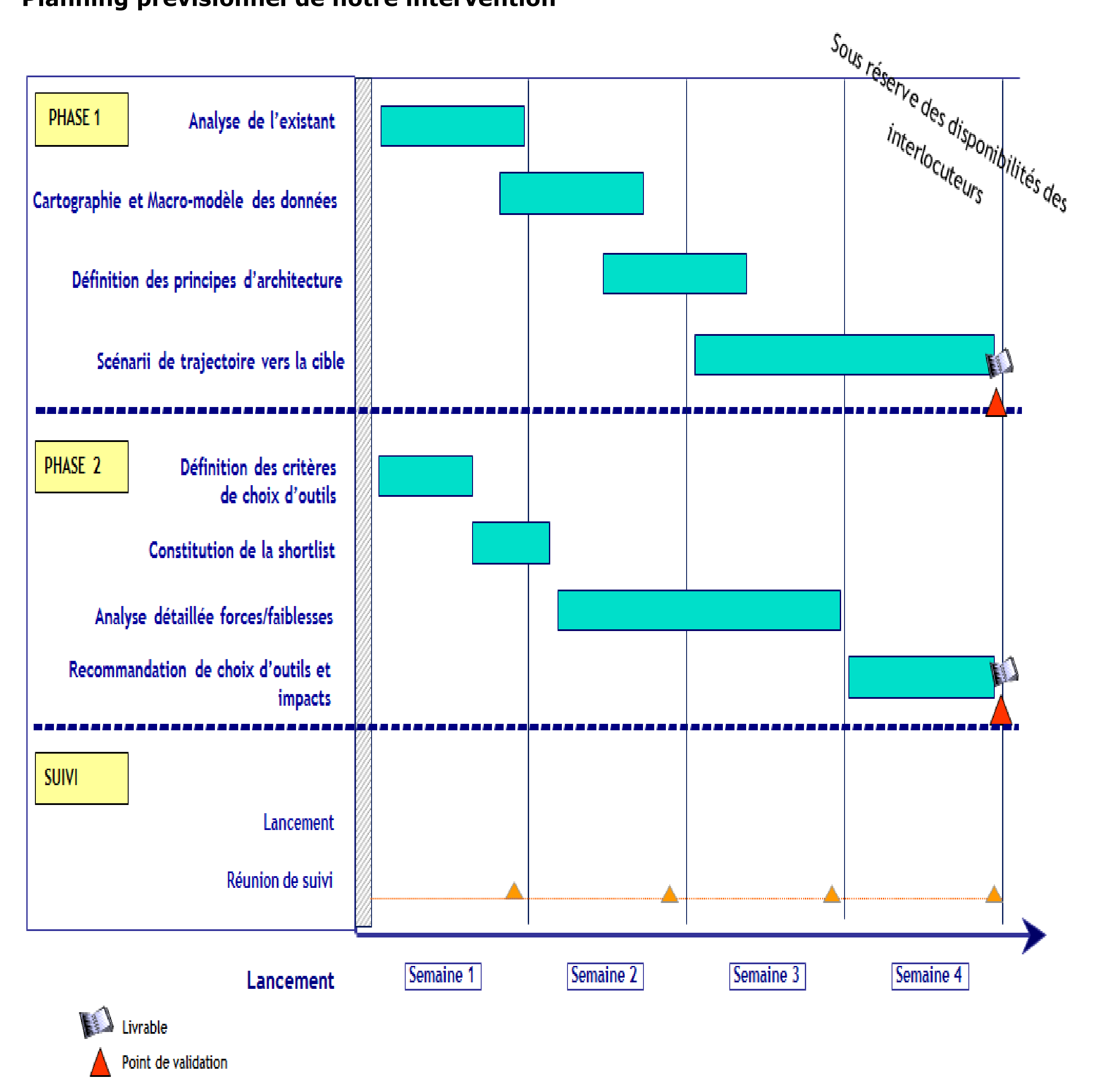
Et la maintenance logicielle

• **CMMI** peut être utilisé pour la maintenance quotidienne des logiciels. Il existe cependant d'autres modèles pour ce type d'activité, comme le modèle de la maturité de la maintenance **S3M**.

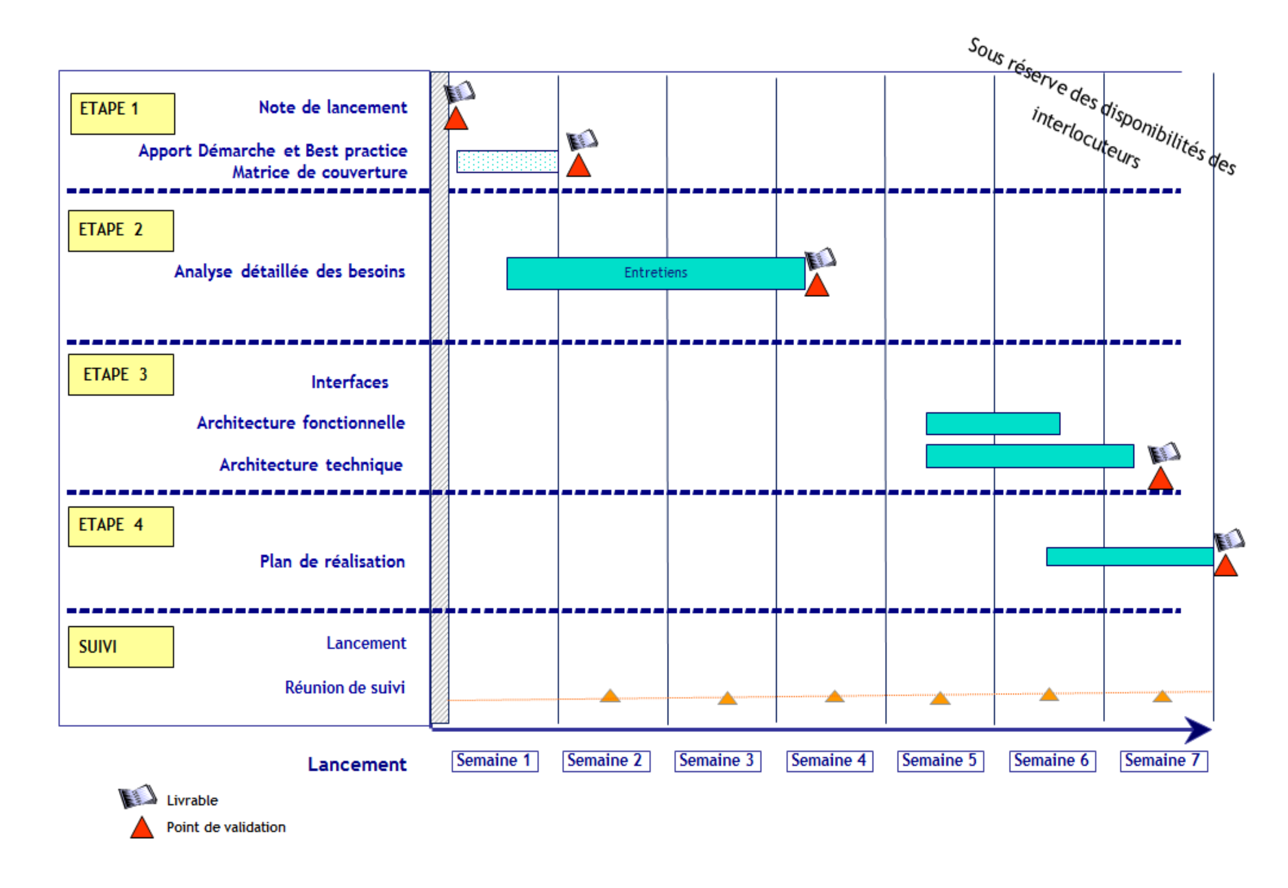
LA PLANIFICATION

Un planning ou un macro planning permet de poser un ensemble de taches sur une période de temps. Il existe des outils spécialisés, mais Excel et PowerPoint ont l'avantage d'avoir un rendu plus visuel et sont sur tous les postes de travail. Bien entendu, tout dépend des exigences de vos clients.

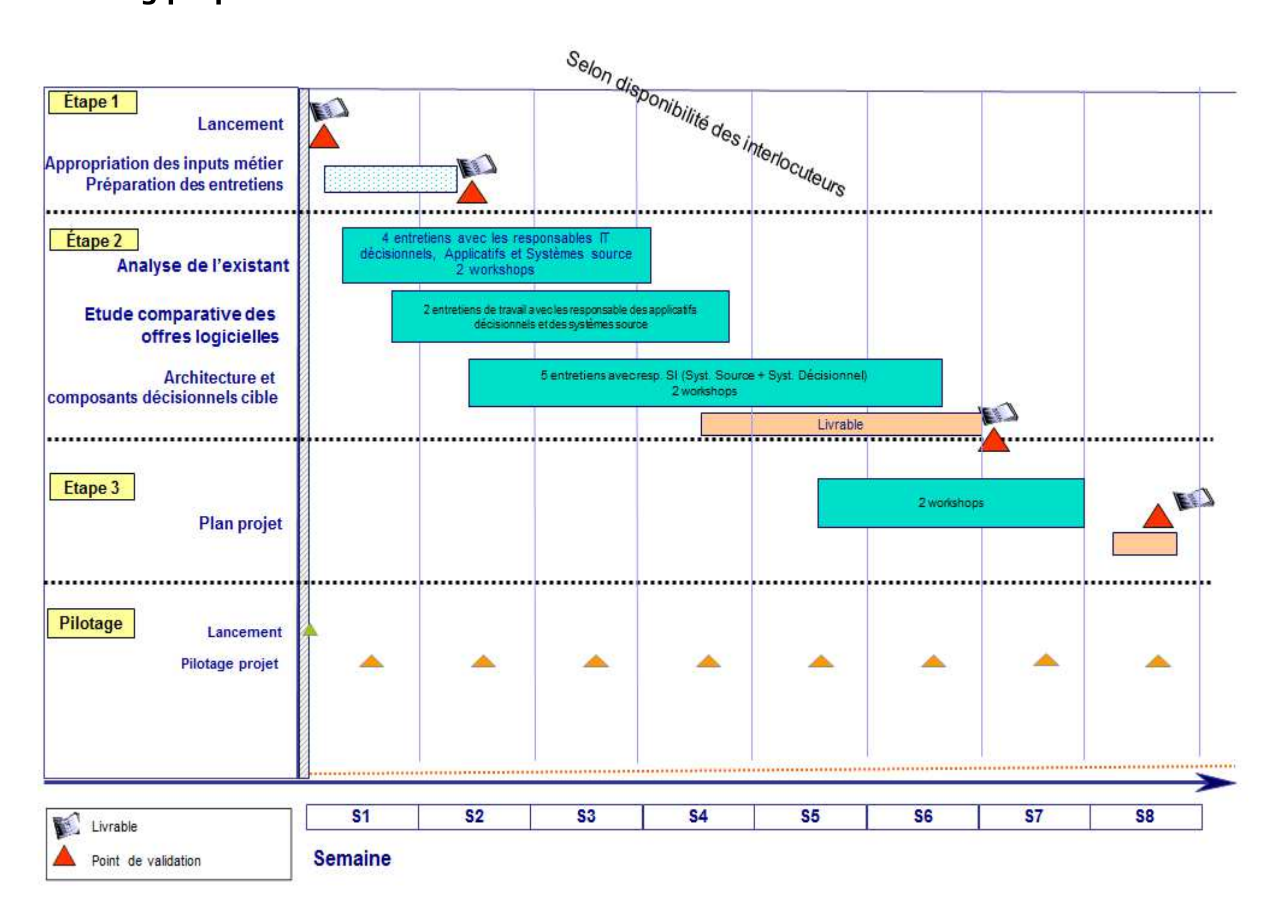
Planning prévisionnel de notre intervention



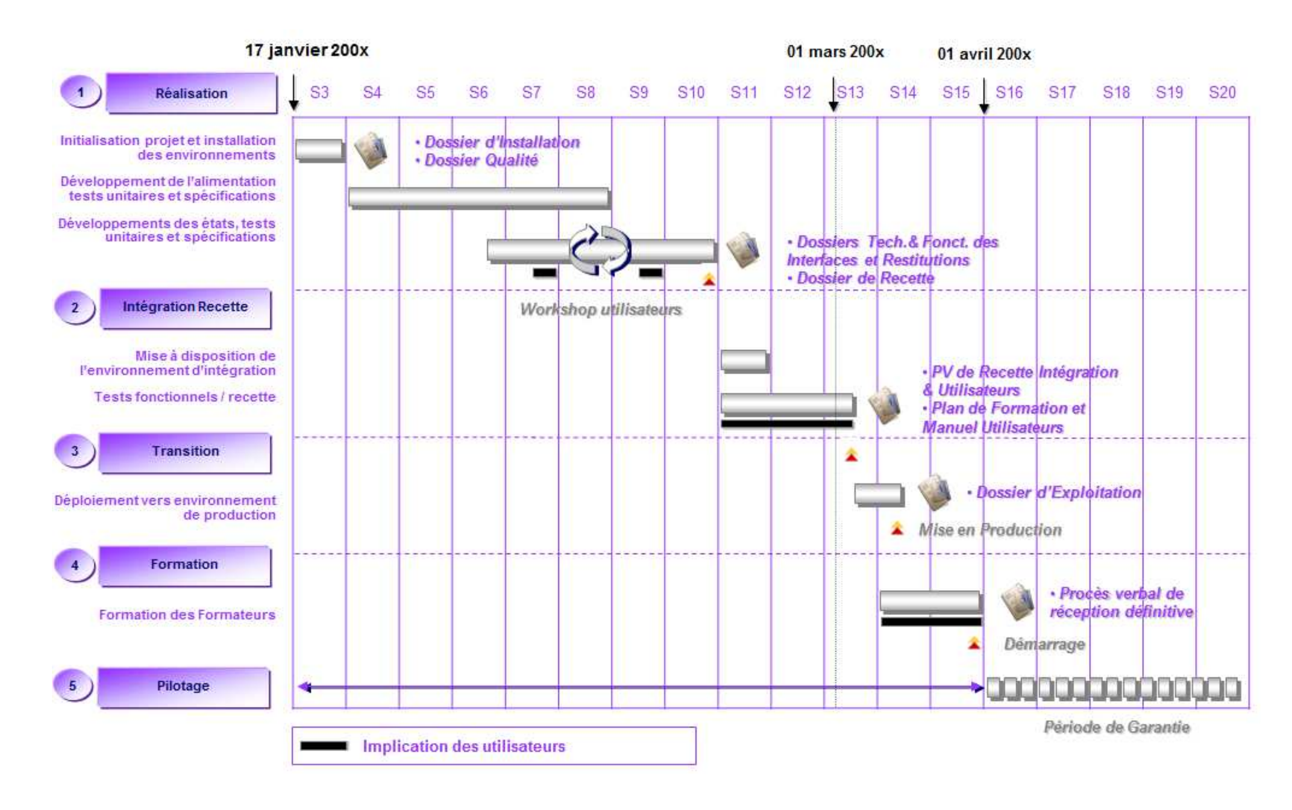
Macro planning conception fonctionnelle et technique détaillée



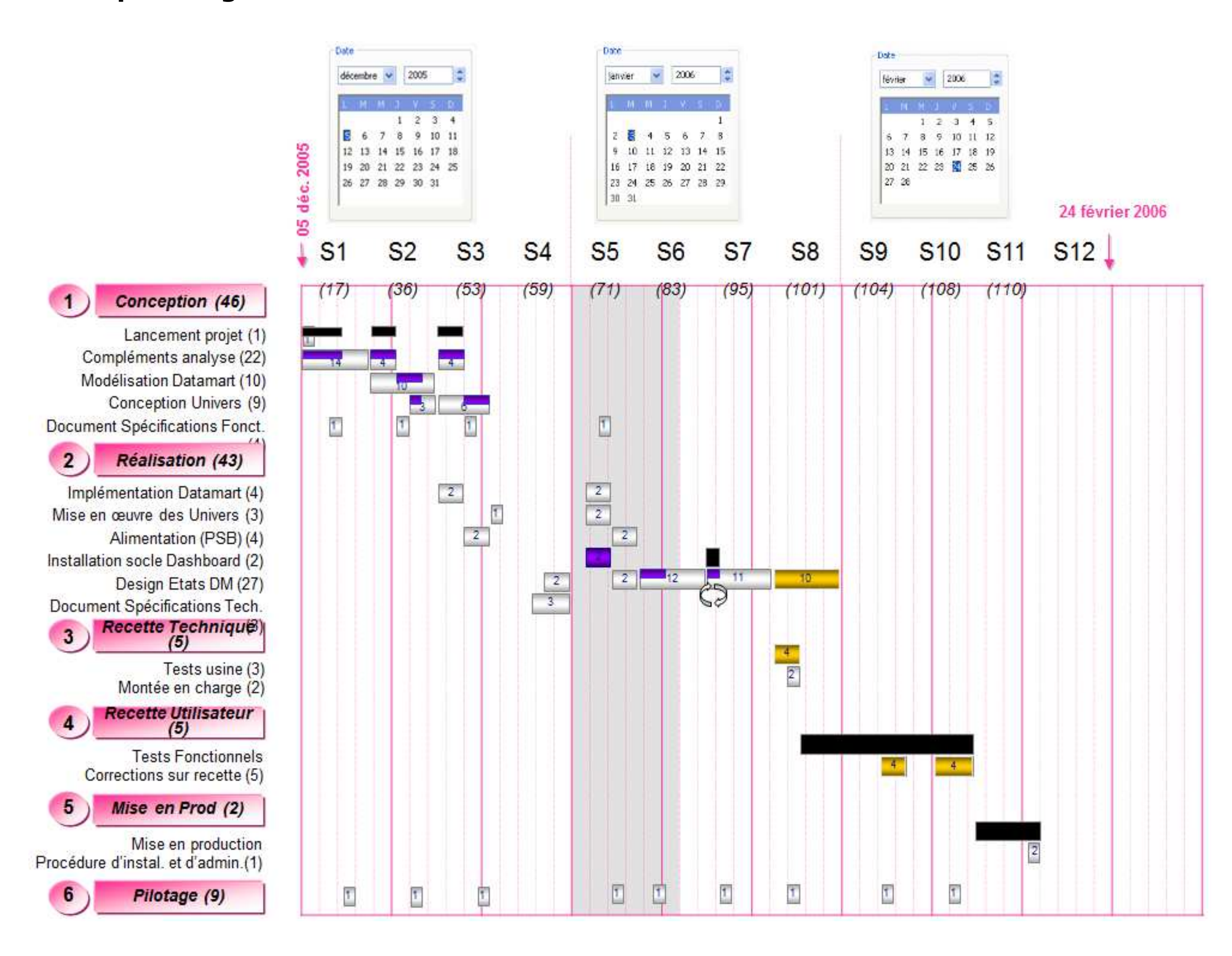
Planning propose



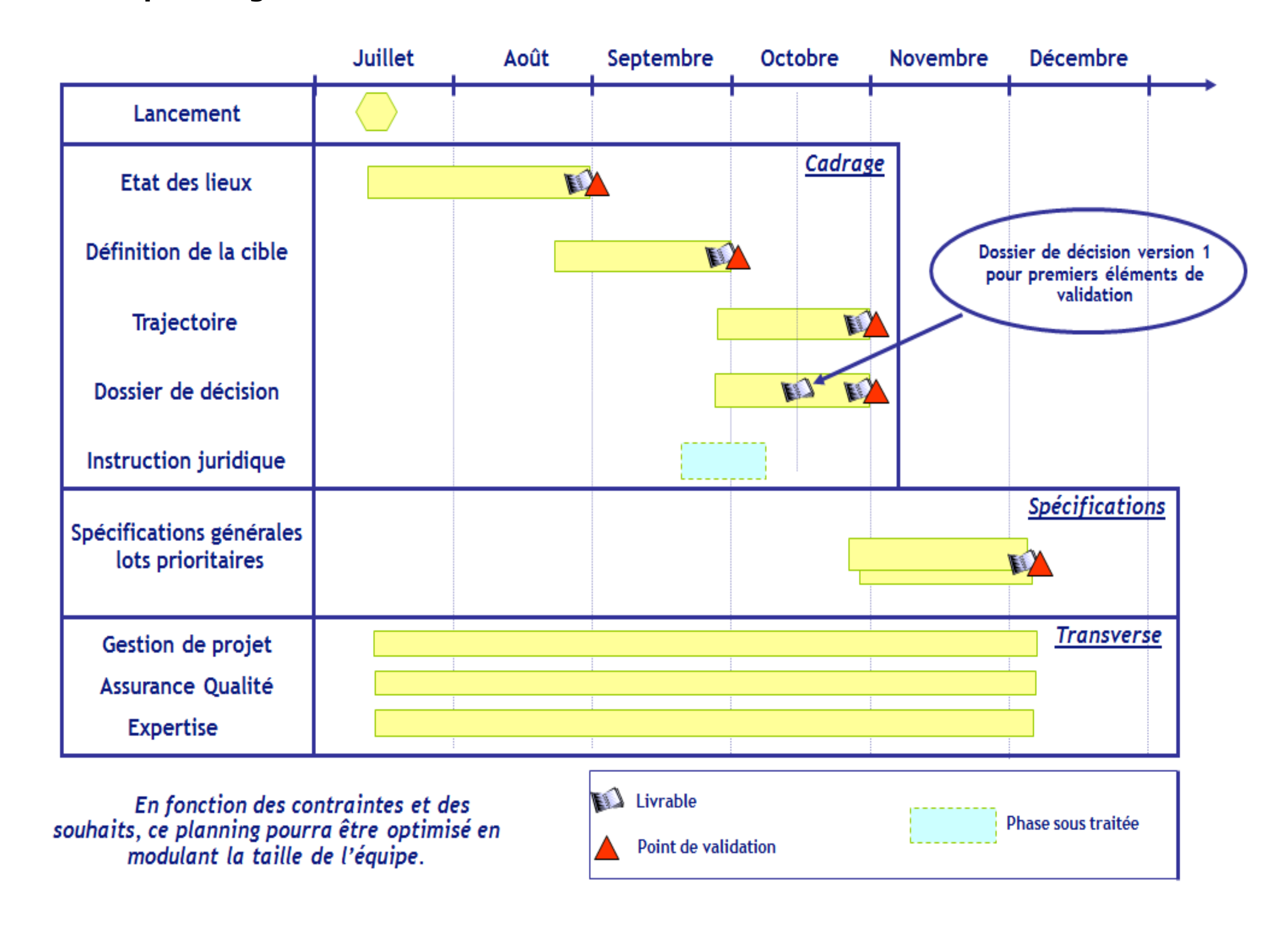
Macro planning de mise en œuvre initial



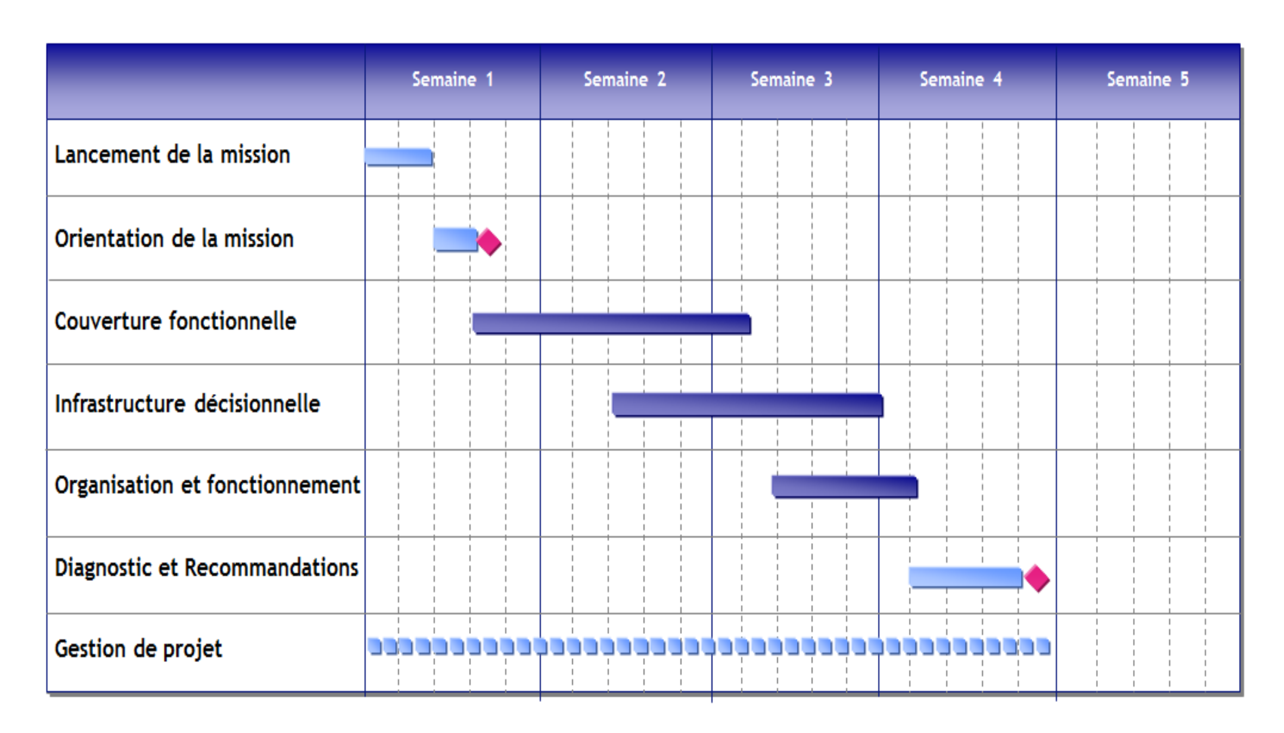
Macro planning



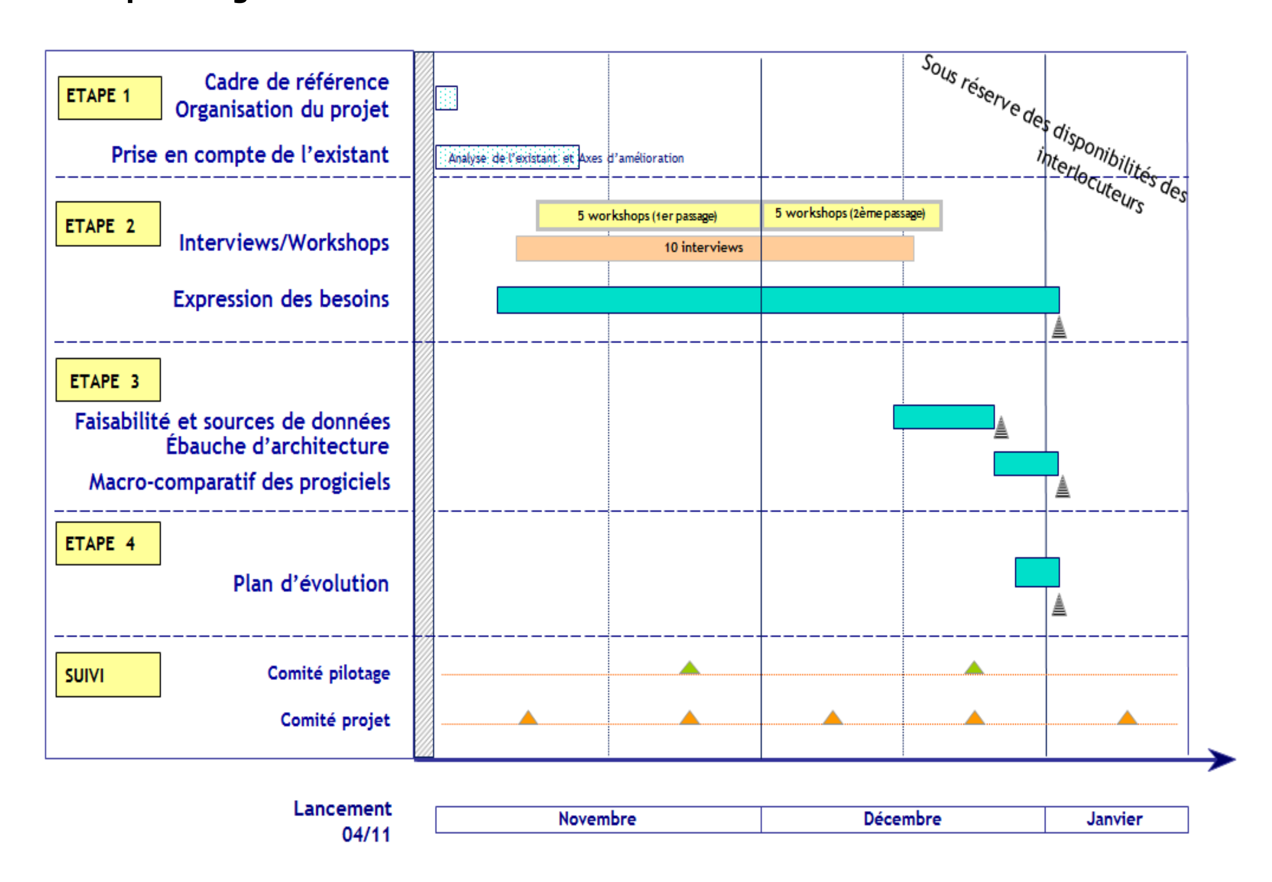
Macro planning



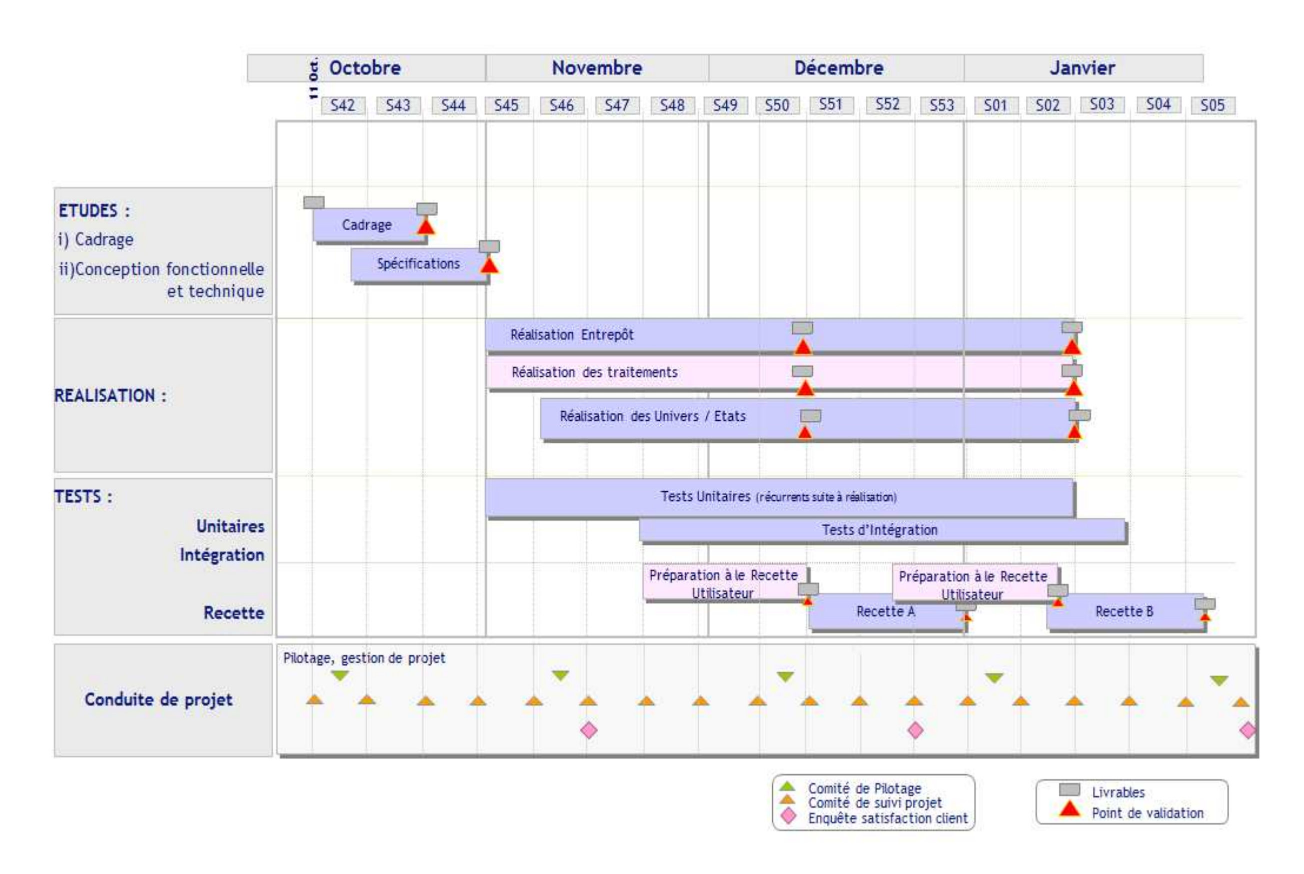
Macro planning



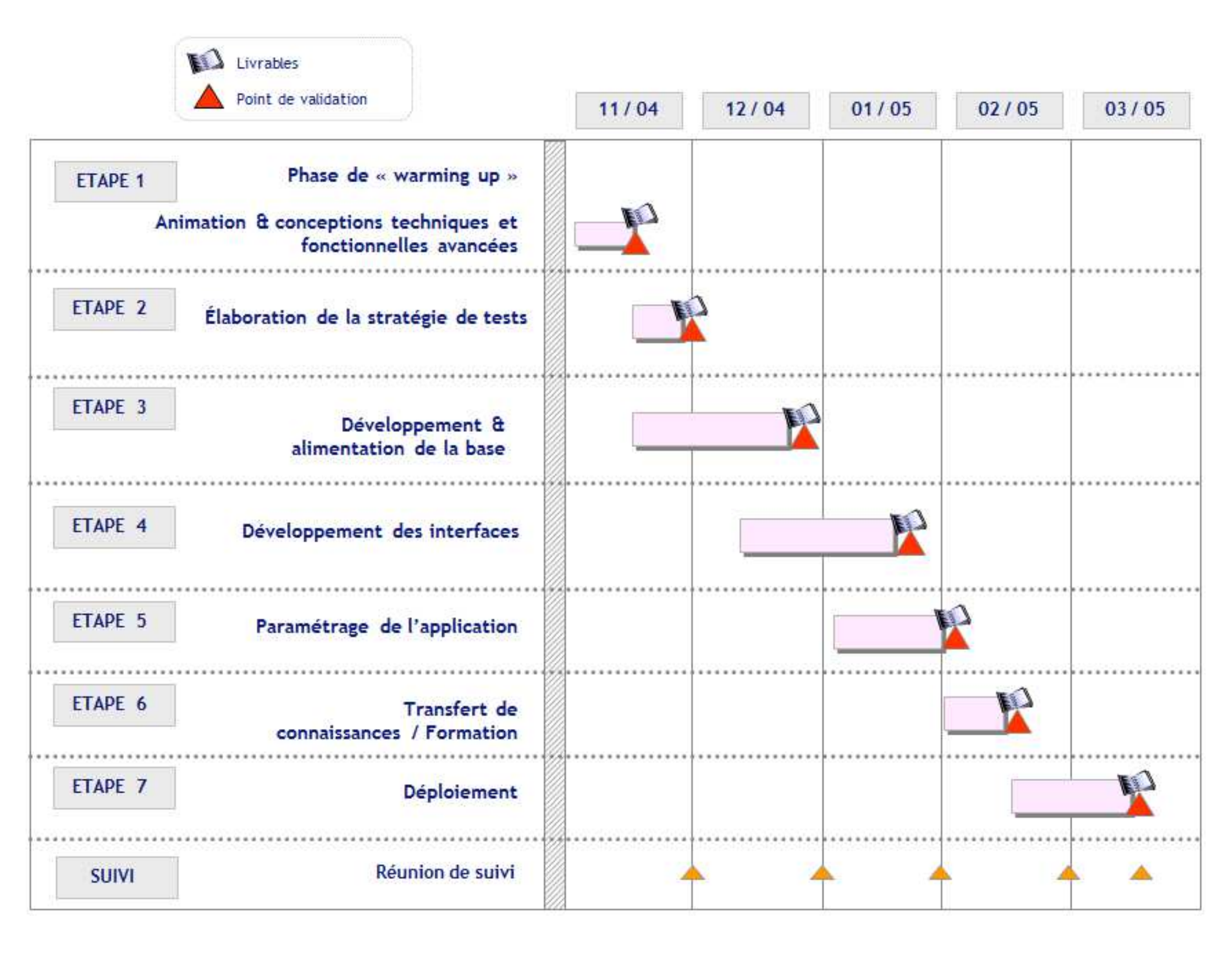
Macro planning

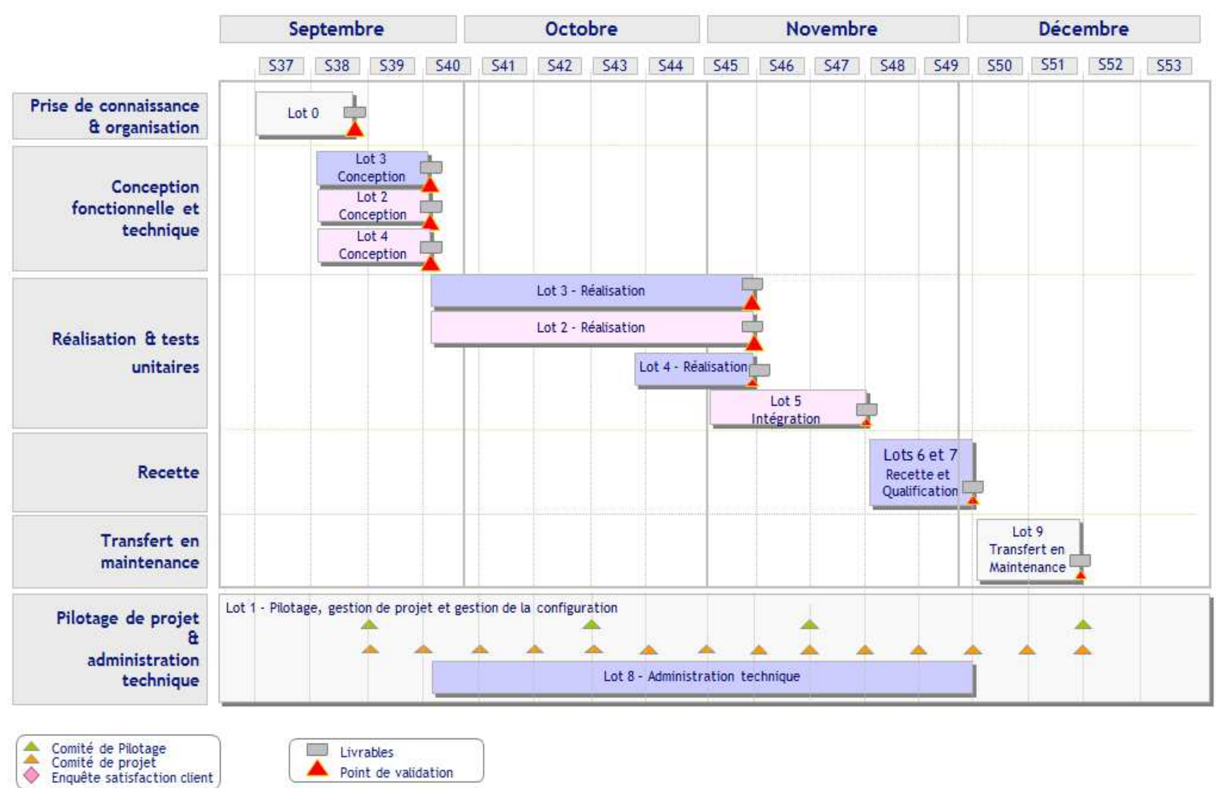


Planning de réalisation et test

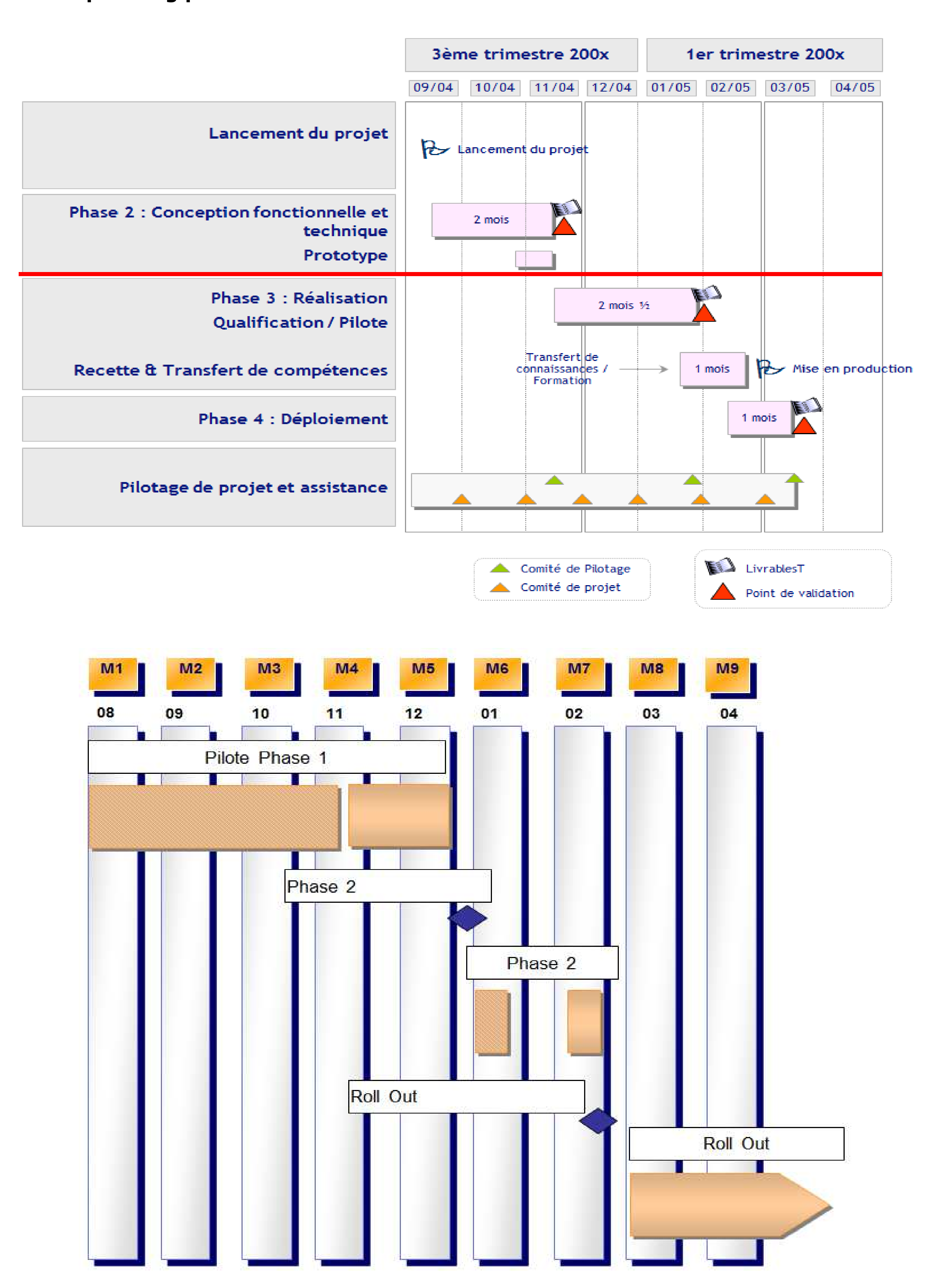


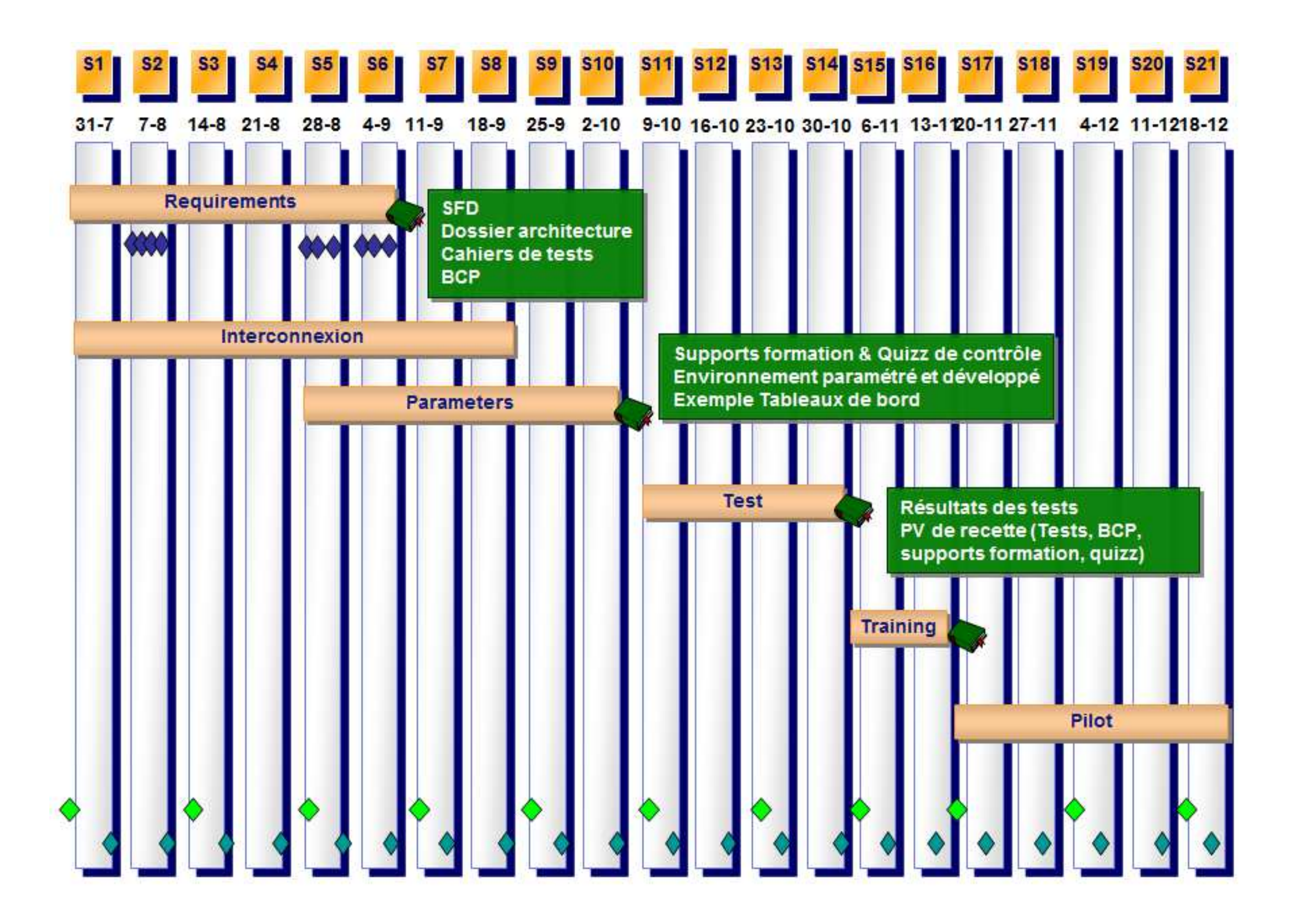
Planning détaillé : Mise en œuvre et déploiement





Macro planning prévisionnel




S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
31-7
7-8
14-8
21-8
28-8
4-9
11-9
18-9
25-9
2-10
9-10
16-10
23-10
30-10
6-11
13-11
20-11
27-11
4-12
11-12
18-12
Requirements
SFD
Dossier architecture
Cahiers de tests
BCP
Interconnexion
Supports formation & Quizz de contrôle
Environnement paramétré et développé
Exemple Tableaux de bord
Parameters
Test
Résultats des tests
PV de recette (Tests, BCP, supports formation, quizz)
Training
Pilot

SOLUTIONS DECISIONNELLES

TECHNIQUE ET MAITRISE D'ŒUVRE

Quelle frontière entre la maîtrise d'ouvrage et la maîtrise d'œuvre dans le décisionnel ?

Je pense que selon son parcours, son expérience, l'organisation des sociétés dans lesquels nous sommes ou avons été, SSII ou client final, chaque personne peut répondre d'une manière différente.

Personnellement, je ne sépare pas franchement la maîtrise d'ouvrage et la maîtrise d'œuvre, sauf pour les sociétés à vocation purement conseil.

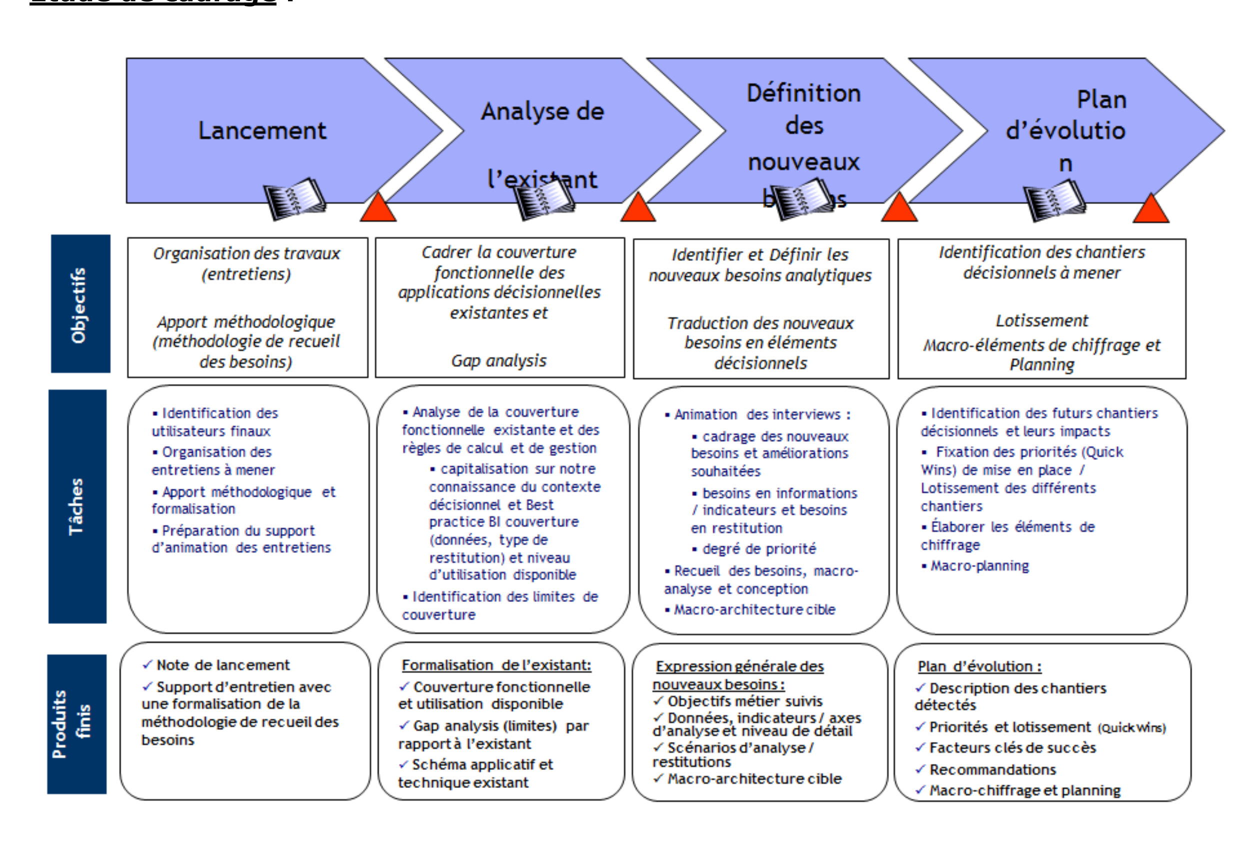
Avec l'expérience, les frontières s'estompent et il est naturel qu'un ingénieur, un chef de projet ou directeur de projet soit amené à tout faire dans le cadre de ses missions.

Je serai donc volontairement large dans mon approche en regard de mon parcours.

PROJET DECISIONNEL

Un exemple de projet décisionnel au sens large du terme avec une étude de cadrage résumée par un schéma.

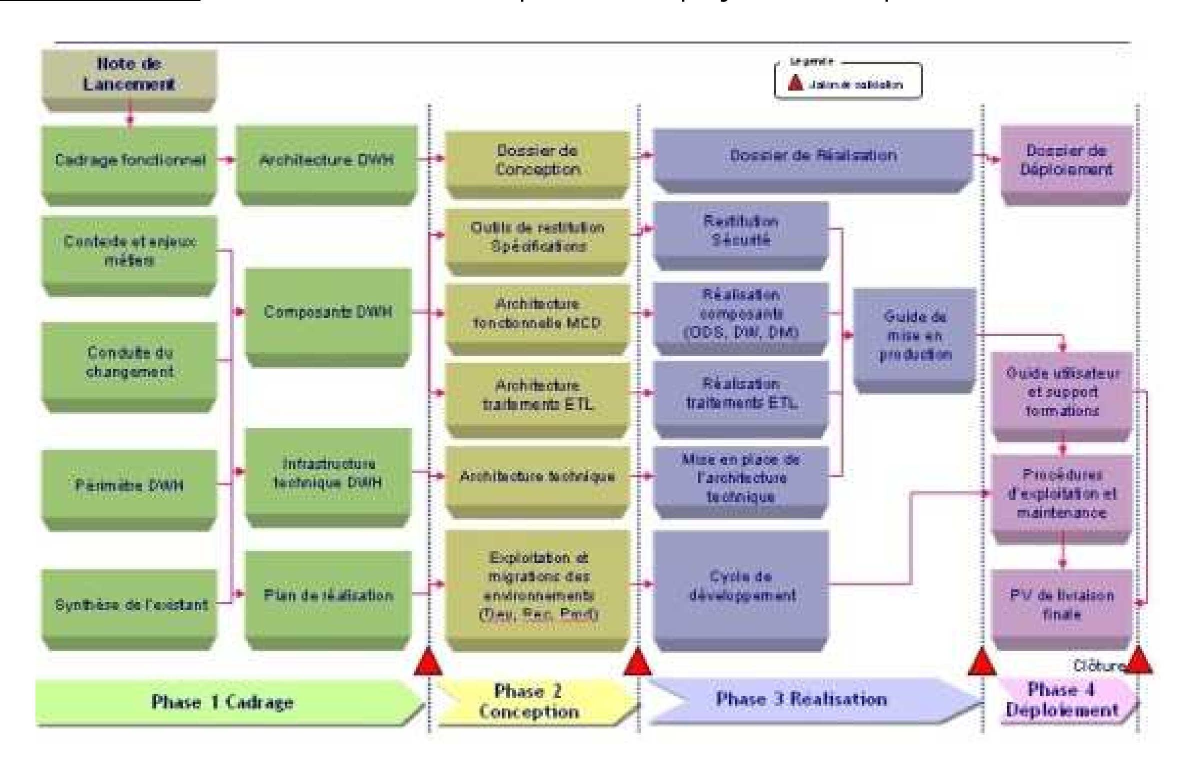
<u>Étude de cadrage</u> :



L'étude de cadrage des besoins en matière de pilotage doit permettre de :

- Recueillir et définir les orientations pour la mise en place d'un système de partage de l'information.
- Cadrer la couverture fonctionnelle des applications existantes.
- Identifier et définir les nouveaux besoins.
- Disposer d'une projection dans le temps des futurs chantiers.
- Définir et appliquer une démarche de mise en œuvre partagée par les différents acteurs concernés.

<u>Mise en œuvre</u> : La mise en œuvre des phases d'un projet résumée par un schéma.



<u>Cadrage</u> : Faire le rapprochement avec le schéma précédent.

<u>Conception</u> : Description détaillée et spécifications des rapports standards (contenu, destinataires, règles d'accès, fréquence de mises à jour…), ainsi que définition de l'architecture technique notamment concernant les aspects sécurité réseau.

Précision et spécifications des processus et des règles de gestion des données (gestion des rejets, qualité, processus d'alerte et d'échange d'information…), règles de sécurité et définitions des profils (administrateurs et utilisateurs…).

<u>Réalisation</u> : Préparation des environnements et des infrastructures techniques, codages des programmes d'alimentation et de traitements des données, création des bases de données (ODS, Datawarehouse, Datamart)

Définition de la sécurité, création des environnements de reporting, des rapports standards, publication dans le portail.

Réalisation des tests unitaires, des tests d'intégration et accompagnement des utilisateurs lors de la recette.

<u>Déploiement</u> : Mise en production et déploiement de la solution, accompagnement dans le déploiement.

<u>Transversalement</u> : Assistance des utilisateurs

Animation pour s'assurer de l'adhésion des utilisateurs au projet, démonstration prototype et rédaction d'un support de présentation, identification des utilisateurs clés.

Préparation des formations, rédactions des guides utilisateurs, des scénarios de recette, des jeux de données.

Accompagnement au déploiement, rédaction du support de déploiement et assistance.

<u>Transversalement</u> : Suivi et organisation du projet

Supervision des équipes techniques et fonctionnelles, garantie de la qualité des produits finis, préparation et suivi opérationnel des comités de pilotage.

SPECIFICATIONS

Principes généraux

Indicateurs, KPI : Chaque indicateur clé peut être défini suivant les critères suivants :

• **Son nom.** Dans la mesure du possible, le nom doit être normalisé, c'est-à-dire avoir le même sens pour tout le monde et ne doit pas être sujet à interprétation selon un contexte de restitution ou par le rôle de la personne qui le consulte.

• **La règle de gestion à l'origine de son calcul.** La donnée issue des systèmes de production peut être restituée de manière brute, mais elle peut être le résultat de calculs parfois complexes, et/ou enrichis par des règles de gestions fonctionnelles.

L'utilisateur pour comprendre l'indicateur, doit connaître les règles qui le régissent sans pour autant connaître le métier de celui qui l'a créée.

• **La source de données correspondante.** Plusieurs sources sont possibles pour un indicateur parfois consolidé et agrégé dans une base de données. La donnée est alors anonymisée. Connaître la ou les sources lui redonne du sens.

• **Sa fréquence de rafraîchissement.** La plupart des indicateurs peuvent généralement être rafraîchis sur tous les éléments de l'axe temps, et conservés selon une durée adéquate.

Périodes jour, semaine mois, trimestre année pour les périodes les plus courantes.

La fréquence de rafraîchissement est plus liée aux besoins utilisateurs qu'a la disponibilité même journalière des données en provenance des systèmes de production.

• **Sa période d'observation.** Observation jour, semaine mois, trimestre année pour les périodes les plus courantes et comparatifs de la donnée entre des périodes de l'axe temps.

D'autre part, si un indicateur est rafraîchi tous les jours sur la base des informations du jour J-1, ceci ne signifie pas nécessairement qu'il faudra mettre en place un suivi mensuel de celui-ci, parfois les données relatives à un mois donné continuent souvent de bouger une fois ce dernier écoulé.

Peuvent être privilégiés les cumuls à J-1, avec des comparaisons avec le même mois de l'année précédente et les cumuls au 31.12 de l'année précédente.

En l'absence d'objectifs précis, les seules comparaisons seront donc faites par rapport à une donnée passée.

DEFINITION DES INDICATEURS

Un tableau récapitulatif permet d'avoir une vision globale d'un Datawarehouse ou d'un Datamart.

Extrait d'un tableau :

Typologie	Nom de l'indicateur	Origine	Objet correspondant/Règle de calcul	Fréquence d'observation	Période d'observation	Cumul YTD (du début de l'année à J-1)
Commandes - Nombre	Nb de commandes non totalement livrées	Achats	Il s'agit de toutes les commandes pour lesquelles le code livraison finale n'est pas coché pour tous les postes. Peuvent être pluriannuelles.	Journalière	Tous exercices (photo à un instant t)	Non
	Nb de commandes appel sur contrats nationaux	Achats	Commandes en référence à un contrat cadre national, pour lesquelles « Appel sur contrat » de la classe « Commandes »=O et « Type de contrat (local/national) » de la classe « Contrats cadres »=ZN	Journalière	Mensuelle (Mois courant jusqu'à J-1)	Oui
	Nb de commandes saisine a priori	Achats	Nb de commandes dont le montant est supérieur à un seuil. Des seuils différents sont définis en fonction du segment d'achat (prestations intellectuelles, commandes de travaux, prestations de service, fournitures). Ces segments d'achat sont des regroupements de Groupes de Marchandises spécifiques par rapport à cet indicateur. Un flag indique permet de détecter les marchés devant être soumis à la CCM (« Marché niveau CCM »=1 de la classe « Commandes »)	Journalière	Mensuelle (Mois courant jusqu'à J-1)	Oui
Commandes – Montants	Montant des commandes non totalement livrées	Achat	Il s'agit du montant de toutes les commandes pour lesquelles le code livraison finale n'est pas coché pour tous les postes. Peuvent être pluriannuelles.	Journalière	Tous exercices (photo à un instant t)	Non
	Montant facturé	Achats	« Montant facturé » de la sous-classe « Détail des factures » de la classe « Postes de commande »	Journalière	Mensuelle (Mois courant jusqu'à J-1)	Oui
Contrats cadre	Nombre de contrats cadre locaux valides	Achats	Comptage des contrats cadre dont « Type de contrat (local/national) de la classe « Contrats cadres » =ZL et pour lesquels la date du jour est inférieure ou égale à la date de fin de contrat cadre (« Fin de contrat cadre date »)	Journalière	Tous exercices (photo à un instant t)	Non
Fournisseurs	Nb de fournisseurs	Achats	« Nb de fournisseurs actifs (SIREN) » de la classe	Journalière	Mensuelle (Mois	Oui

Matrice de ventilation des indicateurs

L'objectif de cette matrice est de donner un aperçu des critères de ventilation des indicateurs ciblés et par conséquent des fonctionnalités demandées en termes de reporting, états, navigation multidimensionnelle au travers des données (croisement d'axes) et d'accès à l'information de détail.

Pour chacun des indicateurs décrits dans le paragraphe précédent, une grille permet d'associer à chaque indicateur les critères de ventilations souhaitables parmi la liste suivante :

Attention : Ceci ne veut pas dire que dans les analyses du tableau de bord tous les critères mentionnés par rapport à un indicateur donné pourront être appliqués en même temps.

Note : Il n'a y pas une totale cohérence entre la grille ci-dessous et la liste du dessus. J'ai procédé à une réduction aléatoire des lignes, voir l'intégralité du document n'amenait rien de plus à notre exemple.

Indicateurs	UNITE	PROJET	EOTP	ACTIVITE	CENTRE COUT	NAT DE DEP	DOM ACHAT	SOUS DOM ACH	GPE DE MARCH	COMPTE CG	TYPE CDE	COMMANDE	POSTE CDE	FOURNISSEUR
Nb de commandes non totalement livrées	X	X	X	X	X	X	X	X	X	X	X			X
Nb de commandes éditées	X	X	X	X	X	X	X	X	X	X	X			X
Nb de commandes en régularisation	X	X	X	X	X						X			X
Nb de commandes binômées	X	X	X	X	X		X	X			X			X
Nb de commandes appel sur contrats locaux	X	X	X	X	X		X	X			X			X
Nb de commandes appel sur contrats nationaux	X	X	X	X	X		X	X			X			X
Nb de commandes saisine a priori	X	X		X			X	X			X			X
Montant des commandes non totalement livrées	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Montant commandé	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Montant des commandes en régularisation	X	X	X	X	X						X			X
Montant des commandes binômées	X	X	X	X	X		X	X			X			X
Montant commandé appel sur contrats locaux	X	X	X	X	X		X	X			X			X
Montant commandé appel sur contrats nationaux	X	X	X	X	X		X	X			X			X
Montant facturé	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Montant livré	X	X	X	X	X	X	X	X	X	X	X	X	X	X

PHILOSOPHIE GENERALE DES RESTITUTIONS

Des données agrégées aux données détaillées

Le besoin d’information d’un tableau de bord de pilotage peut être défini par rapport au champ d’intervention de son destinataire :

• **Stratégique :** Le besoin d’information porte alors sur les effets à long terme de la stratégie et sur la pertinence des orientations stratégiques au regard des finalités ; la nature de l’information est plus synthétique et les indicateurs de type composites.

• **Opérationnel :** Le besoin d’information dépend de la capacité à agir sur les sous-systèmes. L’information est alors de nature analytique, centrée sur les variables qui déterminent les performances à court terme de l’entité. Elle doit permettre une rétroaction sur le processus de production.

La vision instantanée d’indicateurs clés de performance (KPI) est essentielle pour une prise de décision rapide. Il s’agit d’un pré requis qui, en épargnant aux décideurs une immersion dans des masses de données transactionnelles, leur permet d’identifier les faits les plus cruciaux pour l’activité.

Parallèlement, cette interface doit servir de radar de détection des anomalies, intégrant idéalement une série de seuils dont le dépassement est notifié sur-le-champ aux décideurs.

Mais aussi importants soient-ils pour une gestion efficace du processus fonctionnel, ces indicateurs clés ne représentent que le sommet de la pyramide. L’amélioration de la performance du processus fonctionnel ne réside pas seulement dans une vue synthétique des informations, mais également dans la possibilité d’examiner les données de détail qui découlent de ces instantanés.

En d'autres termes, l'infrastructure idéale doit être conçue pour détecter les problèmes au niveau le plus fin, en agrégeant et en explorant les données détaillées de chaque maillon du processus.

Une vision en temps réel et unique des données

Dans ce contexte, il est important de disposer d'une vision en temps réel des données. Le manque d'actualité des informations est d'ailleurs un défaut fréquemment reproché aux applications et aux bases de données décisionnelles.

Mais ces données peuvent être mouvantes. Tous les jours des réaffectations de montants peuvent se répercuter sur les mois précédents, ce qui rend obsolète d'un jour sur l'autre tout pré calcul d'un indicateur sur une base mensuelle pour l'année en cours et même sur l'année précédente. Il est parfois inutile de figer la donnée, si ce n'est en fin d'année.

Ceci rend également difficiles l'affectation et le suivi d'objectifs spécifiques par rapport aux indicateurs.

Il faut alors privilégier les cumuls en temps réel depuis le début de l'année aux analyses mensuelles, en ayant en point de mire la valeur cumulée au 3112 de l'année précédente.

Quoi qu'il en soit, du fait des croisements multicritères s'appliquant à la plupart des indicateurs, il n'est forcément envisageable de pré calculé ces derniers au croisement de tous ces axes d'analyse à la fois.

En l'absence d'une différenciation des indicateurs par profil utilisateur (indicateurs composites pour la direction versus indicateurs directs pour les opérationnels) et du fait d'un besoin exprimé de navigation au travers des données vers le haut (données agrégées) tout comme vers le bas (données de détail) à tous les niveaux de l'organisation, les modes de restitutions proposés vont être communs à tous les utilisateurs de l'application.

MODES DE RESTITUTION PROPOSES

Des données synthétiques en point d'entrée

Une vision synthétique des indicateurs clés devra être fournie en point d'entrée de l'application cible et de chacune des rubriques proposées. Les indicateurs visualisés seront les mêmes pour tous les utilisateurs, mais seront déclinés par rapport à l'unité de rattachement de chaque utilisateur si les dimensions de l'application doivent être sécurisées.

De ce fait, le Directeur visualisera un montant commandé en cumulé depuis le début de l'année relatif à son pôle, alors que le responsable d'un département visualisera le montant commandé cumulé de son propre département.

L'agencement de ces indicateurs dans des pages d'accueil sera cependant toujours le même, indépendamment du profil associé.

L'objectif principal de tableaux de bord n'est pas de donner une information de détail, mais d'aider le responsable à identifier rapidement les écarts significatifs de la valeur réelle d'un indicateur par rapport aux valeurs de référence au sein de sa propre unité (pôle, département ou service).

C'est cette notion d'écart qui doit être mise en évidence en priorité sur le tableau de bord. Avant même de connaître la valeur de la donnée, le responsable doit être informé de l'existence ou non d'un écart.

Exemple de pictogramme associé à l'écart (flèche vers le haut ou vers le bas) constituant un premier niveau de lecture :

Exercice en cours (cumuls)

	J-1(A)	M(A-1)		31/12(A-1)
Nb de commandes éditées	200	220	↘	600
Montant commandé	3000	2800	↗	5300
Montant livré	2800	2600	↗	6000
Montant Facturé	2500	2900	↘	5600
Nb de fournisseurs (SIREN)	300	250	↗	670
Nb commandes seuils CCM	20	19	↗	45
Montant cdes seuils CCM	1300	1200	↗	3450

Les principaux écarts seront mesurés :

• Entre un indicateur mensuel et le même indicateur pour le mois précédent.
• Entre un indicateur mensuel cumulé sur l'année en cours et le même indicateur cumulé pour l'année précédente.
• Entre un indicateur mensuel cumulé sur l'année en cours et le même indicateur cumulé au 31.12 de l'année précédente

Bien entendu, cette liste n'est pas limitative, et vous pouvez, selon les besoins exprimés, décliner sans limitation les règles régissant vos indicateurs ainsi que leurs modes de représentation.

L'utilisateur doit ensuite avoir la possibilité s'il le souhaite d'approfondir son niveau d'information au moyen de représentations graphiques ou d'éléments d'information plus détaillés.

Des analyses de détail permettant la navigation au travers des différents axes d'analyse.

Les analyses de détail doivent permettre à partir d'un indicateur et d'un modèle de restitution de naviguer au travers des données suivant les principaux axes d'analyse. Cette navigation peut s'effectuer de trois manières différentes :

• Au travers de filtres permanents
• Au travers de hiérarchies de navigation liées au mode de représentation de l'indicateur (analyse descendante ou ascendante)
• Au travers d'une exploration transverse via des axes d'analyse complémentaires

Ce mode de « recherche » de l'information est très interactif par rapport à une approche de reporting standard. Typiquement lorsque l'on navigue de cette manière au travers des données, on sait d'où l'on part, mais on ne sait pas a priori sur quelles données, à quel niveau de détail et au croisement de quels critères l'analyse va s'arrêter.

Dans une analyse de détail, on distingue donc :

Un ou plusieurs Indicateurs.

Des Filtres permanents (listes déroulantes) permettant d'analyser les données sous différents angles, notamment suivant l'axe Temps (mois, année) et suivant l'axe Unité (Pôle/Direction, Département, Service) ou la notion de Centre de livraison.

Des Hiérarchies permettant à partir du mode de représentation original de descendre vers les données de détail et de remonter (analyse descendante et analyse ascendante)

La possibilité d'utiliser des axes d'exploration complémentaires.

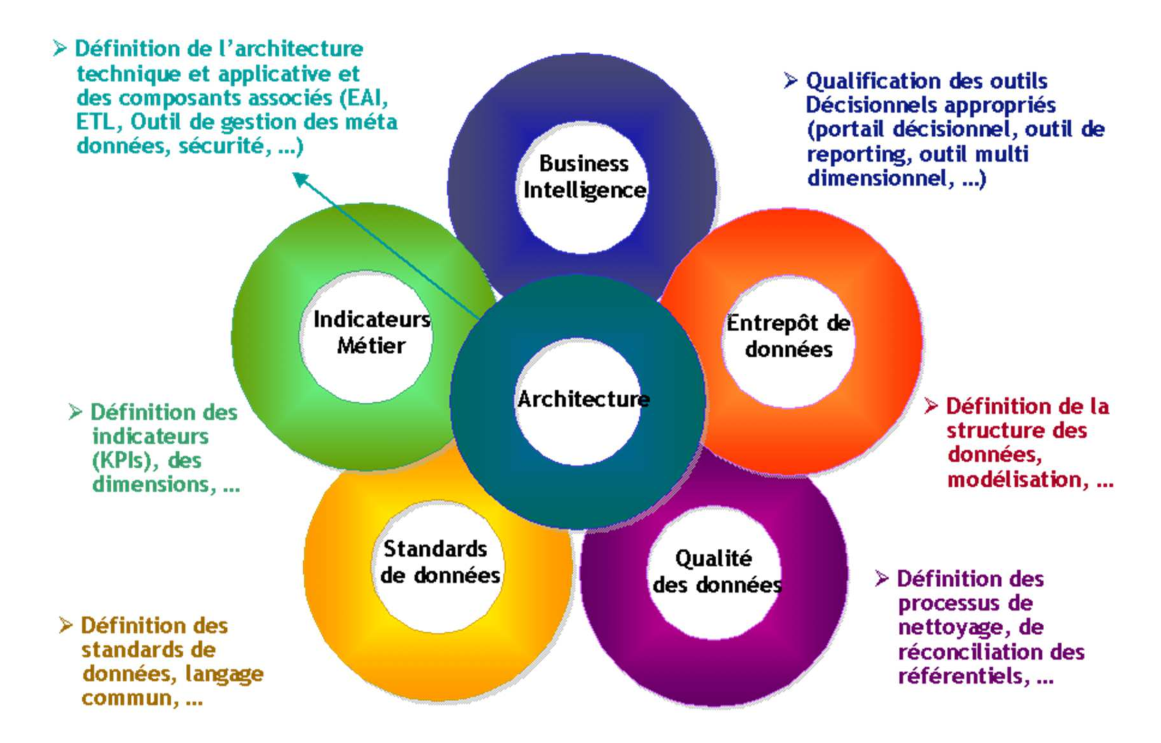
Des Mécanismes de navigation : Permettent à partir du mode de représentation original de descendre vers les données de détail et de remonter (analyse descendante et ascendante) ainsi que d'utiliser des axes d'exploration complémentaires.

FACTEURS DE SUCCES

FACTURES DE SUCCES EN TERME DECISIONNEL

LES CARACTERISTIQUES D'UN PROJET DECISIONNEL

Selon nous, les thèmes importants à aborder dans un projet décisionnel sont représentés sur le schéma ci-dessous :



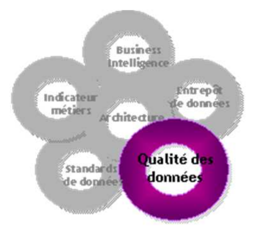


Data Standard (standard de données)

Les objectifs des standards de données sont : obtenir un langage commun sur l'ensemble des définitions des données.

Définition des données (méta données) :

Mode de calcul standard
Convention de nommage et définitions des entités qui ont un impact fonctionnel transverse
Qualification et modification des données
Processus, procédures et règles de gestion



Qualité des données

Sur la base de notre expérience, les objectifs sont : Analyse du niveau de qualité des systèmes sources par échantillonnage, mise en place de stratégies et procédures d'amélioration de la qualité des données, implémentation d'indicateurs de suivi de la qualité.



Business Intelligence (décisionnel)

Qualification et implémentation des outils décisionnels (outils de reporting et analytiques)

Requêtes Ad Hoc et rapports standard (standardisés, préformatés, et planifiés)

On-Line Analytical Processing (OLAP), Multidimensional On-Line Analytical Processing (MOLAP)...

La sélection des outils de reporting et d'analyse est faite sur la base d'une méthodologie interne en respectant les besoins fonctionnels identifiés.



Business Measures (indicateurs métier)

Afin de mesurer des activités transverses dans une organisation globale, les indicateurs de performance peuvent être définis comme suit : les « flux » fonctionnels et processus métiers détermineront quels indicateurs de performance, Standards de données, et besoins analytiques sont requis.

Une fois les indicateurs de performance et les Standards de données, identifiés, la définition détaillée des indicateurs est effectuée. Celle-ci inclut les règles de gestion et modes de calcul appropriés, les données sources sur lesquelles s'appuyer, etc.

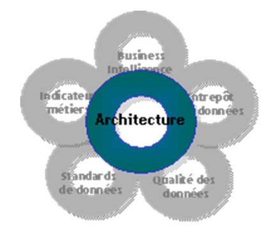


Data Warehouse/Business Warehouse (entrepôts de données, magasins de données et/ou bases de données orientées métier)

Une base de données orientée métier, intégrée, et contenant une collection de données en support à l'aide aux décisions.

Un entrepôt de données cohérentes pouvant être manipulées et analysées de manière intuitive par des utilisateurs finaux.

Une base de données informationnelle utilisée pour mettre des informations en provenance des systèmes de productions.
Modélisation des informations dédiées à l'analyse en respectant les Standards de données.



Architecture

Cartographie applicative (schéma de flux, sources des données…).
Cartographie technique (architecture n-tiers, client-serveur...).
Description des flux, outils, fréquence...

La cartographie technique précise les extractions, transformations, le chargement des données et la gestion des Méta Données. Figureront aussi les sources de données en support des composants décisionnels.

La sélection des outils de reporting et d'analyse est faite sur la base d'une méthodologie interne en respectant les besoins fonctionnels identifiés.

FACTEURS DE SUCCES ET RISQUES DU PROJET

LES FACTEURS CLES DE SUCCES

Afin de vous aider à atteindre vos objectifs, et au vu des contraintes spécifiques de votre environnement et de votre projet, nous avons bâti notre proposition, objet du présent document, autour des principes clés suivants :

Une **méthodologie éprouvée de cadrage décisionnel fondée sur une approche pragmatique :** forts de nos expériences de missions de cadrage décisionnel et de projets d'intégration, nous avons synthétisé notre savoir-faire au sein d'offres qui traitent les aspects d'alimentation, de mise en cohérence, de fiabilisation et de stockage des données ainsi que de la production, la définition et la restitution des indicateurs.

L'implication permanente des utilisateurs afin de :

- Concerner et informer l'ensemble des utilisateurs de tous les groupes métiers concernés ;
- Favoriser le transfert de compétences ;
- Favoriser l'appropriation des produits ;
- Définir une organisation pérenne, connue et reconnue par l'ensemble des interlocuteurs ;
- Favoriser la mise en œuvre des solutions les plus adaptées aux besoins exprimés.

La mise en place d'une structure de pilotage forte afin :

- D'assurer la réalisation de chaque tâche dans les délais impartis ;
- De suivre les indicateurs nécessaires à la vision globale du projet ;
- De maîtriser le changement et sa perception par les différents utilisateurs.

L'implication d'un membre de notre comité de direction en charge de l'entité Business Intelligence dans le cadre de ce projet.

La mise en place des prototypes en phase de spécification a été prévue **pour :**

- Concrétiser la solution auprès des utilisateurs ;
- Augmenter la visibilité des utilisateurs en cours de projet ;
- Valider les hypothèses fonctionnelles sur la base d'une application tangible ;
- Valider l'architecture technique et applicative de la solution définie ;
- Adopter une démarche pragmatique, proche du besoin exprimé ;
- La démarche itérative dans la mise en œuvre de solutions qui permette d'éviter « l'effet tunnel »

Notre expérience des projets se traduit par notre certification qualité. Les outils mis en place dans le cadre de cette certification étant garants d'une excellente traçabilité des événements et d'une très bonne anticipation des risques.

LA MAITRISE DES RISQUES INTRINSEQUES A LA MISE EN PLACE D'UN SYSTEME D'INFORMATION DECIONNEL

La mise en place avec succès d'un système d'information décisionnel implique la maîtrise d'un certain nombre de risques.

Risque	Réponse de BT CSI
Les fonctionnalités demandées évoluent sans cesse et ne sont jamais stabilisées.	Un cadrage rigoureux du périmètre, une conduite du changement facilitant la prise de décision. Validation systématique des choix et des livrables.
La participation des utilisateurs à la définition des besoins est insuffisante.	L'identification des utilisateurs clés et leur implication rapide dans le projet. Le suivi des charges et de l'implication de chacun.
Les données issues des systèmes amont ne sont pas fiables.	Une approche qualitative qui permet d'identifier les données insuffisamment fiables et de conseiller les travaux à mener pour les améliorer.
Des spécifications en effet tunnel, tout mettre « au cas où ».	L'expression claire des besoins et le cadrage précis du périmètre, la mise en lumière des facteurs coûts, la prise en compte de la qualité des données. La proposition de solutions alternatives.
La méthodologie manque de rigueur.	Une méthodologie éprouvée et rigoureuse, une anticipation des risques permettant de qualifier au plus vite la faisabilité des demandes exprimées.
Les utilisateurs ne se sentent pas impliqués.	La planification de réunions régulières d'information, et de validation. La proposition d'une démarche itérative dans la mise en œuvre de solutions qui permette d'éviter « l'effet tunnel ».
La solution sélectionnée ne répond pas aux besoins fonctionnels.	La connaissance des solutions du marché et des éditeurs, une approche fonctionnelle et technique de l'outil, la capacité à déterminer rapidement la faisabilité des demandes par rapport aux possibilités de l'outil.

LOGICIEL DECISIONNEL

BUSINESS OBJECTS

Je ne vais pas expliquer ce que fait **BusinessObjects** (SAP), vous ne le liriez pas, par contre, techniquement, il y a beaucoup de choses à dire.

C'est une technologie qui évolue vite, trop vite selon moi, il est très difficile de se tenir à jour, et nous finissons par nous y perdre.

Comme le sujet technique est vaste, le meilleur moyen pour débuter est de commencer par l'installation.

Sur Windows, je ne pense pas que vous vous heurtiez à de grandes difficultés, par contre sur Unix ou Linux, cela peut sembler plus difficile.

Je ne suis en aucun cas un spécialiste de ces systèmes, je tâtonne seulement, cependant, j'ai été amené à l'installer dans le cadre de missions, voici donc une procédure pour vous aider, et en fait, c'est assez simple.

INSTALLATION BUSINESS OBJECTS SUR LINUX.

Pré requis

Avant de créer la base de données relationnelle à intégrer à **BusinessObjects Enterprise**, prenez en compte les sections ci-après qui décrivent en détail les paramètres requis à la création de la base de données et les paramètres à tester avant de commencer l'installation de **BusinessObjects Enterprise**. Une exigence est valable pour tous les types de bases de données : la base de données relationnelle doit être configurée pour utiliser un codage de caractères Unicode (par exemple UTF-8).

Note : Configuration requise pour la base de données Sybase. Lorsque vous créez la base de données pour le CMS, vous devez vérifier que la taille de page correspond à 8 Ko.

Remarque : La taille de page par défaut de la base de données Sybase est égale à 2 Ko, ce qui est insuffisant pour le CMS. Pour une exécution optimale du CMS, la taille de page doit être égale à 8 Ko. La taille de page est configurée au cours de la création de la base de données et il est impossible de la modifier par la suite.

Préparation du serveur de base de données existant

Vous ou l'administrateur de base de données devez préparer le serveur de base de données de manière à permettre au CMS de s'y connecter. Cette opération doit être effectuée avant d'installer **BusinessObjects Enterprise**.

Au cours de l'installation, le programme vous demandera si vous souhaitez installer MySQL ou utiliser une base de données existante. Si vous spécifiez l'utilisation d'une base de données existante, vous devrez indiquer des détails concernant la base de données.

Bien que vous deviez indiquer les détails concernant votre base de données pendant l'installation de BO, il ne vous sera pas demandé de fournir son nom, sauf si vous utilisez une version existante de MySQL

Pour préparer votre base de données :

1. Créez une base de données relationnelle vide sur votre serveur de base de données : ex: BOE115

2. Créez un login « **bobje** » (identifiant/mot de passe) de connexion ayant par défaut votre base de données BO (ici BOE115) et attribuez-lui un mot de passe sécurisé.

Vérifiez que ce nouvel utilisateur dispose des droits nécessaires pour créer, modifier, supprimer des tables et créez des procédures afin que **BusinessObjects Enterprise** puisse, si nécessaire, modifier la base de données.

Création et vérification de la connectivité à la base de données du CMS

Pour créer des tables et écrire des données dans votre nouvelle base de données CMS, les scripts d'installation BO doivent établir une connexion au serveur de base de données. En d'autres termes, lorsque vous vous connectez à Linux avec le nom d'utilisateur avec lequel vous allez effectuer l'installation, l'environnement Shell par défaut doit inclure les variables d'environnement de base de données et/ou les fichiers d'initialisation appropriés. Dans le cas contraire, le programme d'installation ne pourra pas accéder à la base de données CMS à l'aide du logiciel client de votre base de données.

Les variables d'environnement et/ou les fichiers requis par les scripts d'installation dépendent du type de serveur de base de données que vous exécutez.

D'autres variables d'environnement de base de données doivent être définies afin que le script d'installation puisse utiliser correctement le logiciel client de base de données. Avant d'exécuter le script d'installation, testez l'environnement Shell du compte utilisateur sous lequel vous allez installer **BusinessObjects Enterprise** afin de vérifier la connexion à la base de données et les droits.

Testez la connexion à la base de données en vous connectant, puis tapez une commande de create table et de drop table pour vous assurer que tout fonctionne.

Présentation de l'installation

BusinessObjects Enterprise fournit une architecture ouverte et souple qui prend en charge une multitude de scénarios de déploiement et de configuration. Avant d'installer **BusinessObjects Enterprise**, vous devez :

• Passer en revue votre système pour vous assurer qu'il correspond à la configuration de base requise pour une installation de **BusinessObjects Enterprise**.
• Vérifier que tous les ordinateurs qui font partie du déploiement **BusinessObjects Enterprise** communiquent correctement entre eux./dd>
• Déterminer l'emplacement d'installation des composants.
• Choisir une méthode d'installation.

Configuration système requise

Veuillez noter que les composants suivants doivent être généralement installés et configurés correctement avant toute installation de **BusinessObjects Enterprise** :

• Serveur d'applications Web (sauf si vous installez Tomcat en même temps que **BusinessObjects Enterprise**)
• Logiciel de base de données compatible avec les bases de données CMS et d'audit (à moins d'installer MySQL pendant l'installation de **BusinessObjects Enterprise**)
• Pour en savoir plus sur la configuration matérielle requise, reportez-vous au fichier Platforms.txt fourni avec votre distribution du produit, ou au fichier PDF relatif aux plateformes prises en charge disponible sur le site de support technique de Business Objects.

Autorisations Linux

Pour effectuer une installation utilisateur ou système sous Linux, le compte d'utilisateur employé pour exécuter l'installation doit disposer d'autorisations de lecture, d'écriture et d'exécution sur le répertoire où **BusinessObjects Enterprise** sera installé. Il n'est pas nécessaire de disposer de droits root pour effectuer une installation utilisateur ou système de **BusinessObjects Enterprise**. En fait, si vous tentez une installation avec ces droits, elle échouera.

Nom d'hôte et configuration réseau requis

Votre serveur Linux doit disposer d'un nom d'hôte spécifique avant d'exécuter le script d'installation. Pour définir ou modifier cette information sur votre système, vous devez disposer des droits root requis. Si vous n'êtes pas familiarisé avec ces procédures, consultez la documentation de votre système Linux.

Lorsque vous installez **BusinessObjects Enterprise** sur plusieurs ordinateurs, assurez-vous que chacun d'entre eux peut communiquer via TCP/IP avec l'ordinateur sur lequel est installé votre CMS (Central Management Server).

Création d'un compte, d'un répertoire de base et d'un environnement de connexion

Créez un compte d'utilisateur et un groupe spécifiques sous lequel exécuter les traitements **BusinessObjects Enterprise** en arrière-plan. Pour terminer l'installation, vous devrez vous connecter sous ce nom d'utilisateur.

Bien que des droits root soient nécessaires à la configuration de ce compte, ce type de droits n'est pas requis pour le compte en lui-même. Il n'est pas nécessaire que les scripts d'installation ou **BusinessObjects Enterprise** soient exécutés en tant que root.

Suivez vos procédures administratives habituelles pour effectuer les tâches recommandées.

Pour créer un compte en vue de l'installation de BusinessObjects Enterprise :

1. Créez un groupe ou utilisez un groupe existant. Créez un compte utilisateur, puis définissez le nouveau groupe comme groupe principal de cet utilisateur. Attribuez un mot de passe sécurisé au nouveau compte d'utilisateur.

2. Créez le répertoire à l'emplacement où vous souhaitez installer **BusinessObjects Enterprise**. Par défaut, le répertoire actuel, à savoir le répertoire dans lequel vous exécutez

install.sh, sera utilisé comme répertoire de base pour l'installation. Vous pouvez remplacer ce paramètre par défaut et utiliser le répertoire de votre choix au moment de l'installation. Vous verrez alors le répertoire que vous spécifiez pour le répertoire d'installation, indiqué sous la forme **REP_INSTALL** dans ce document.

3. Assurez-vous que le compte que vous avez créé dispose des droits de lecture, d'écriture et d'exécution sur le nouveau répertoire de base **HOME**.

4. Attribuez un Shell de connexion par défaut au nouvel utilisateur et créez ou modifiez les scripts de connexion appropriés pour le compte d'utilisateurs. En outre, veillez à ce que les scripts de connexion définissent un environnement de connexion par défaut satisfaisant ces conditions requises :

• Toutes les commandes et tous les utilitaires requis par le programme d'installation install doivent être accessibles dans la variable d'environnement **PATH**.
• L'environnement de connexion de l'utilisateur doit configurer l'environnement de la base de données de telle sorte que le programme d'installation install puisse accéder à votre logiciel client de base de données.
• L'environnement de connexion de l'utilisateur doit configurer des paramètres régionaux par défaut pris en charge par votre système Linux et **BusinessObjects Enterprise**.

Vérification des commandes et des utilitaires requis

Pour que le programme d'installation « **install** » s'exécute correctement, les commandes et utilitaires suivants doivent être installés sur votre système Linux :

/bin/sh, pwd, read, touch, uname, expr, hostname, sed, awk, chown, grep, tail, tar, id, dirname, gzip, stty, ulimit, which

Ces commandes et utilitaires standard sont normalement disponibles sur la plupart des distributions Linux. Ces utilitaires doivent être accessibles dans la variable d'environnement PATH du compte d'utilisateur que vous utilisez lors de l'installation de **BusinessObjects Enterprise**.

Définition des paramètres régionaux

Avant d'installer **BusinessObjects Enterprise**, configurez votre système d'exploitation de manière à utiliser des paramètres régionaux pris en charge par **BusinessObjects Enterprise XI** pour le type et la version de votre système Linux. Pour que votre système d'exploitation utilise les paramètres régionaux adéquats à chaque exécution de **BusinessObjects Enterprise**, attribuez aux variables d'environnement **LC_ALL** et **LANG** les paramètres régionaux correspondant à votre environnement de connexion. (Par exemple, si vous utilisez un Shell C, définissez ces variables d'environnement dans le fichier .login.)

Astuce : Saisissez locale pour vérifier que toutes les variables d'environnement connexes des paramètres régionaux (telles que **LC_MONETARY**, **LC_NUMERIC**...) ont été correctement définis par **LC_ALL**.

INSTALLATION BUSINESSOBJECTS

Distribution du produit directement à partir d'un CD

Par défaut, le répertoire en cours, à savoir le répertoire à partir duquel vous exécutez install.sh, sera utilisé comme répertoire de base pour l'installation. Si vous exécutez install.sh sans copier les fichiers dans un emplacement temporaire, vous serez invité à spécifier un emplacement temporaire pour l'installation. Une fois l'emplacement temporaire indiqué, les opérations ci-dessous seront effectuées :

• Les fichiers d'installation seront copiés dans l'emplacement temporaire en question.
• Le programme d'installation sera fermé.

Vous devrez alors accéder à l'emplacement temporaire que vous avez spécifié, puis exécuter install.sh à partir de cet emplacement.

Copie de la distribution du produit sur l'ordinateur

Par défaut, le répertoire en cours, à savoir le répertoire à partir duquel vous exécutez install.sh, sera utilisé comme répertoire de base pour l'installation. Vous pouvez copier la distribution du produit dans un répertoire sur l'ordinateur, puis exécuter install.sh à partir de cet emplacement. Cette option présente l'avantage de ne pas devoir indiquer un emplacement temporaire pour placer les fichiers lorsque vous exécutez install.sh.

Pour copier la distribution du produit sur l'ordinateur :

• Connectez-vous à votre système UNIX sous le nouveau compte désigner pour l'installation de BusinessObject Enterprise.
• Copiez les fichiers d'installation de la distribution du produit dans un répertoire temporaire en utilisant la commande ci-dessous, où /mnt/cd est associé au lecteur de CD et tmp représente un répertoire temporaire où vous souhaitez stocker les fichiers d'installation : /mnt/cd/install -t /tmp/

Répétez cette procédure pour chaque disque contenu dans la distribution du produit.

1. Montez le périphérique contenant les fichiers d'installation.

2. Saisissez ./install.sh

Remarque : Si vous exécutez install.sh sans copier les fichiers dans un emplacement temporaire, vous serez invité à spécifier un emplacement temporaire pour l'installation. Voir "**Distribution du produit directement à partir d'un CD**", pour savoir pourquoi cette opération est requise et comment procéder.

3. Sélectionnez la langue d'installation, puis appuyez sur Entrée.



4. Prenez connaissance du contrat de licence.

Tapez « **y** » pour accepter les clauses du contrat de licence et poursuivre l'installation.

5. Indiquez si vous souhaitez installer uniquement **BusinessObjects Enterprise**

Ou à la fois **BusinessObjects Enterprise** et les produits de pilotage des performances.

Pour installer uniquement **BusinessObjects Enterprise**, saisissez le code clé d'activation de **BusinessObjects Enterprise** dans le premier champ Code clé du produit affiché à l'écran, puis appuyez sur Entrée.

Pour installer **BusinessObjects Enterprise** et les produits de pilotage des performances :

Saisissez le code clé d'activation de **BusinessObjects Enterprise** dans le premier champ Code clé du produit affiché à l'écran. Tapez « **x** » dans le champ « **Installer les produits de pilotage des performances** », puis appuyez sur la touche **TAB** pour quitter le champ. Le champ « **Code clé du produit** » sera effacé. Saisissez la clé du Dashboard Manager pour le « **Code clé d'activation des produits de pilotage des performances** » dans le deuxième champ « **Code clé** » du produit affiché à l'écran.



Remarque : Le code clé que vous saisissez ici n'active que le produit correspondant. Pour activer d'autres produits de pilotage des performances, vous devez saisir les clés de licence de ces produits dans la Central Management Console (CMC), une fois l'installation terminée. Pour en savoir plus sur cette tâche, voir le Guide d'administration de **BusinessObjects Enterprise**.

6. Sélectionnez l'emplacement d'installation du produit.

Pour accepter le répertoire d'installation par défaut (votre répertoire actif), appuyez sur Entrée. Pour créer votre propre répertoire, appuyez sur la touche Retour arrière pour effacer le répertoire en cours et le remplacer par le chemin d'accès à votre répertoire d'installation.



7. Sélectionnez le type d'installation à effectuer.

Vous avez le choix entre les options d'installation Utilisateur et Système.

Lorsque vous choisissez une installation utilisateur, tous les composants requis sont installés.

Lorsque vous choisissez l'installation système, tous les composants requis sont également installés, mais le programme d'installation crée en plus un script d'initiation au niveau du système. Ce script crée des entrées dans les niveaux d'exécution du système d'exploitation qui démarrent les serveurs **BusinessObjects Enterprise** lorsque le serveur Linux est démarré et arrêtent ces mêmes serveurs **BusinessObjects Enterprise** lorsqu'un ordinateur est éteint.

Remarque : Pour exécuter l'installation système, vous devez vous connecter à l'aide d'un compte normal. Après l'installation, cependant, vous devez disposer des droits d'accès root pour exécuter le script setupinit.sh. Ce script copie BobjEnterprise115 dans le répertoire /sbin/rc#.

8. Vous allez maintenant être invité à choisir le type d'installation à effectuer.

Vous avez le choix entre cinq types d'installation : nouvelle, étendue, personnalisée, client et silencieuse.

Choisissez une « Nouvelle installation ». Effectuer une nouvelle installation est la façon la plus simple de déployer **BusinessObjects Enterprise**, car tous les composants requis sont installés par défaut sur l'ordinateur.



9. Indiquez si vous souhaitez utiliser une base de données existante ou installer MySQL.

Si vous souhaitez utiliser une base de données existante :

a. Sélectionnez Utiliser une base de données existante, puis appuyez sur Entrée.
b. Sélectionnez le type de base de données pour votre CMS. Vous avez le choix entre MySQL, Oracle, DB2 et Sybase.



c. Renseignez les champs concernant votre base de données, puis appuyez sur Entrée.



<u>Remarque</u> : Vérifier que le numéro de port par défaut du CMS 6400 n’est pas associé à un autre service ou choisir un autre numéro de port libre.

Nom du service Sybase = DataserveurID utilisateur / Mot de passe = Compte BO pour sybaseServeur de noms local = Nom du système serveurNuméro de port CMS = par défaut 6400

d. Si vous souhaitez activer une base de données d'audit, sélectionner oui.
e. Fournissez les informations concernant votre nouvelle base de données d'audit (Nom de la base de données, Numéro de port de la base de données, ID utilisateur, Mot de passe). Décidez si vous souhaitez réinitialiser la base de données. La réinitialisation de la base de données **BusinessObjects Enterprise** effacera la totalité de son contenu.

<u>Remarque</u> : Si vous utilisez une base de données existante, vous devez indiquer la source de sa variable d'environnement afin que le CMS puisse y accéder après un redémarrage du système. Procédez pour cela de l'une des deux manières suivantes :

• Un utilisateur disposant des droits root peut modifier le script BobjEnterprise115 de **BusinessObjects Enterprise** et ajouter la commande qui permet d'indiquer la source de votre base de données. Ce script se trouve à l'emplacement suivant : **REPINSTALLATION"/bobje/init/BobjEnterprise115**.

Cette méthode indiquera la source de la variable d'environnement de la base de données pour tous les utilisateurs.

• Chaque utilisateur peut modifier son propre profil et ajouter la commande qui permet d'indiquer la source de son environnement de base de données. Cette opération doit être effectuée par chaque utilisateur.

10. Choisissez entre l'utilisation d'un Java Application Server existant et l'installation de Tomcat.

Pour installer Tomcat :

a. Assurez-vous que l'option Installer Tomcat est en surbrillance, puis appuyez sur Entrée.
b. Saisissez les ports HTTP, de redirection et d'arrêt du serveur Tomcat, ou appuyez sur Entrée pour accepter les valeurs par défaut.



11. Appuyez sur Entrée pour démarrer l'installation.



Une fois la nouvelle installation terminée, le programme d'installation démarre les serveurs sous forme de démons, puis active chaque serveur enregistré auprès du CMS. Pour contrôler les serveurs manuellement, utilisez le script ccm.sh. Vous devez à présent déployer InfoView et les modules de **BusinessObjects Enterprise**. Si vous n'installez pas Tomcat au cours de l'installation de **BusinessObjects Enterprise**, ces composants doivent être configurés et déployés avant d'être utilisés.

Fin de l'installation système

Si vous avez choisi d'effectuer une installation système, vous êtes invité, une fois cette dernière terminée, à exécuter le script BobjEnterprise115. Le script BobjEnterprise115 copie les scripts de contrôle d'exécution dans les répertoires /sbin/rc#. Une fois implémentés, ces scripts démarrent/arrêtent les serveurs de **BusinessObjects Enterprise** au démarrage/à l'arrêt du système.

Si vous avez choisi une installation utilisateur, Il faut s'assurer que les serveurs BusinessObjects sont lancés après chaque redémarrage du système.

Consignes d'exploitation

Propriétés	Valeurs
Nom de l'hôte (hostname)	Demo
Adresse IP	10.220.10.150
Système d'exploitation	Red Hat Enterprise Linux AS release 4 (Nahant Update 4) : Kernel 2.6.9-42.ELsmp
Protocoles réseau pris en Charge	TCP/IP

Oracle

Tableau 3 : Caractéristiques de configuration Sybase

Propriétés	Valeurs
Version du logiciel	Oracle 10g
Repertoire d'installation	/export/home/miremont/sgbdr/oracle10/
Compte oracle	Login : Mot de passe :
Base de données du CMS	BOE115
Base de données d'audit	(O/N)

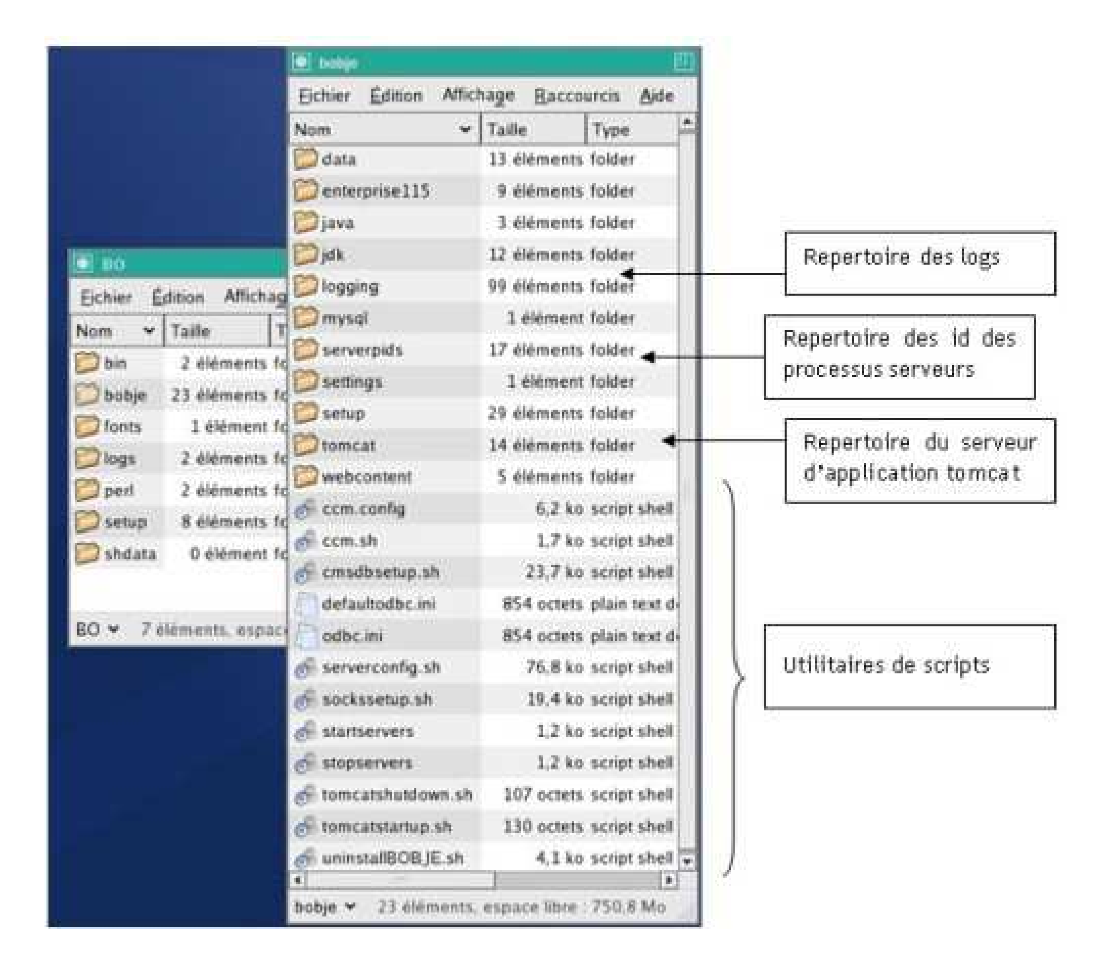
BusinessObjects

Tableau 4 : Caractéristiques de configuration BusinessObjects

Propriétés	Valeurs
Version du logiciel	BusinessOjects XI R2
Repertoire d'installation	/export/logiciels/boxi/
Compte BO	Login : boxi Mot de passe : **********
Type de Base de données du CMS	ORACLE
Nom du service	ORCLSECU
Compte BO pour Oracle	ID utilisateur : dboBOE115 Mot de passe : ***********
Numéro port CMS	6400 (port par défaut)
Serveur de déploiement	Tomcat 5.0.27
JDK	Java(TM) 2 SDK, Standard Edition Version 1.4.2

Pour avoir le détail des paramètres d'installation, éditez le fichier ccm.config situé dans le répertoire « **bobje** » du répertoire d'installation de **BO** (Répertoire d'installation de BO, Type et langue d'installation, type, nom du service et numéro de port de la base de données du référentiel : CMS, …).

Structure de fichiers BusinessObjects



Le répertoire « **bobje** » de **BusinessObjects** contient l'ensemble des fichiers les plus importants (les scripts de démarrage et arrêt des serveurs, le répertoire des logs, les répertoires des id des processus serveurs…).

1 - Liste des outils Linux

La distribution Linux de **BusinessObjects Enterprise** inclut un certain nombre de scripts qui, ensemble, vous offrent toutes les options de configuration. Un certain nombre d'autres scripts vous fournissent d'autres options spécifiques à Linux ou vous servent de modèles pour vos propres scripts. Il existe également plusieurs scripts secondaires qui sont utilisés par **BusinessObjects Enterprise**. Chaque script est décrit ci-dessous et est accompagné d'options de ligne de commande, le cas échéant.

2 - Utilitaires de script

Cette section décrit les scripts administratifs qui vous aident à travailler avec **BusinessObjects Enterprise,** sous Linux. Cette section décrit les concepts qui sous-tendent chacune des tâches que vous pouvez effectuer avec ces scripts.

ccm.sh

Le script ccm.sh se trouve dans le répertoire « **bobje** » de votre installation. Ce script vous fournit une version de ligne de commande du CCM. Quelques exemples d'utilisation.

Exemple :

Ces deux commandes démarrent et activent tous les serveurs. Le CMS (Central Management Server) démarre sur la machine locale et le port par défaut (6400) :

ccm.sh -start all

ccm.sh -enable all

Ces deux commandes démarrent et activent tous les serveurs. Le CMS démarre sur le port 6701, plutôt que sur le port par défaut:

ccm.sh -start all

ccm.sh -enable all -cms MACHINE01:6701

ccm.config

Ce fichier de configuration définit les chaînes de démarrage du serveur et les autres valeurs utilisées par le CCM lorsque vous exécutez ses commandes. Ce fichier est géré par le CCM lui-même et par les autres utilitaires de script de **BusinessObjects Enterprise**. En général, vous ne modifiez ce fichier que lorsque vous devez modifier une ligne de commande d'un serveur. Consulter ce fichier pour avoir des détails sur les paramètres de l'installation.

cmsdbsetup.sh

Le script cmsdbsetup.sh se trouve dans le répertoire « **bobje** » de votre installation. Ce script fournit un programme texte qui vous permet de configurer la base de données CMS, les clusters de CMS et de créer une base de données d'audit.

Vous pouvez ajouter un CMS à un cluster en sélectionnant une nouvelle source de données pour sa base de données CMS. Vous pouvez également supprimer et recréer (réinitialiser) une base de données de CMS, copier des données d'une autre source de données ou modifier le nom de cluster existant.

Exemple : **# ./cmsdbsetup.sh**

Remarque : Avant d'exécuter ce script, sauvegardez votre base de données de CMS en cours. N'oubliez pas de consulter le Guide d'administration de **BusinessObjects Enterprise** pour obtenir davantage d'informations sur les clusters de CMS et sur la configuration de la base de données du CMS. Le script vous invitera à entrer un nom pour votre CMS. Par défaut, le nom du CMS est nomhôte.cms. C'est-à-dire que le nom par défaut d'un CMS installé sur une machine appelée **MACHINE01** est **MACHINE01.cms**. Pour vérifier le nom de votre CMS (ou de tout autre serveur), affichez le contenu de ccm.config et recherchez la chaîne de démarrage du serveur. Le nom en cours du serveur apparaît après l'option -name.

serverconfig.sh

Le script serverconfig.sh est installé dans le répertoire « **bobje** » de votre installation. Ce script fournit un programme textuel qui vous permet d'afficher des informations sur les serveurs et d'ajouter et de supprimer des serveurs de votre installation. Ce script ajoute, supprime, modifie et répertorie des informations du fichier ccm.config.

Lorsque vous modifiez un serveur à l'aide de serverconfig.sh, vous pouvez modifier l'emplacement de ses fichiers temporaires. Pour le CMS (Central Management Server), vous pouvez modifier son numéro de port ou activer l'audit. Pour l'Input File Repository Server et l'Output File Repository Server, vous pouvez saisir le répertoire racine.

Pour ajouter, supprimer, modifier et répertorier des serveurs Linux :

1. Accédez au répertoire « **bobje** » de votre installation.

2. Entrez la commande suivante :

./serverconfig.sh

Le script vous propose une liste d'options :

> 1 - Ajouter un serveur
> 2 - Supprimer un serveur
> 3 - Modifier un serveur
> 4 - Répertorier tous les serveurs dans le fichier de configuration.

3. Saisissez le chiffre correspondant à l'action que vous souhaitez effectuer.
4. Si vous ajoutez, supprimez ou modifiez un serveur, donnez au script toutes les informations supplémentaires qu'il demande.
5. Après avoir modifié ou ajouté un serveur, utilisez le CCM pour garantir que le serveur a bien démarré et qu'il est activé.

Astuce : Le script vous invitera à entrer un nom pour votre CMS. Par défaut, le nom du CMS est nomhôte.cms. C'est-à-dire que le nom par défaut d'un CMS installé sur une machine appelée **MACHINE01** est **MACHINE01.cms**. Toutefois, vous pouvez saisir le nom d'hôte dans ce script pour vérifier le nom de votre CMS (ou de tout autre serveur), afficher le contenu de ccm.config et rechercher la chaîne de démarrage du serveur. Le nom en cours du serveur apparaît après l'option -name.

uninstallBOBJE.sh

Le script uninstallBOBJE.sh se trouve dans le répertoire « **bobje** » de votre installation. Ce script supprime tous les fichiers installés pendant l'installation d'origine de **BusinessObjects Enterprise** en exécutant les scripts dans le répertoire bobje/uninstall. N'exécutez pas vous-même les scripts dans le répertoire uninstall : ces scripts suppriment uniquement les fichiers associés à un composant **BusinessObjects Enterprise** unique, ce qui risque de mettre votre système **BusinessObjects Enterprise** dans un rapport incertain.

Modèles de scripts

Ces scripts sont généralement fournis comme modèles, sur lesquels vous pouvez baser vos propres scripts d'automatisation.

startservers

Le script startservers se trouve dans le répertoire « **bobje** » de votre installation. Ce script peut être utilisé comme modèle pour vos propres scripts : il est fourni comme exemple pour montrer comment vous pouvez configurer votre propre script permettant de démarrer les serveurs de **BusinessObjects Enterprise** en exécutant une série de commandes dans le CCM.

Exemple : **# ./startservers/p>**

Cette commande démarre les services dans l'ordre suivant :

1. Central Management Server : Le CMS se charge de gérer une base de données spécifique à votre système **BusinessObjects Enterprise** à laquelle les autres composants peuvent accéder selon leurs besoins. Les données stockées par le CMS concernent les utilisateurs et les groupes, les niveaux de sécurité, le contenu de **BusinessObjects Enterprise** et les serveurs.

2. Page Server : Réponds aux requêtes de page en traitant les rapports et en générant des pages EPF (Encapsulated Page Format). Le Cache Server et le Page Server travaillent en étroite collaboration. Plus précisément, le Page Server répond aux requêtes de page émises par le Cache Server.

3. Cache Server (Crystal Reports ou Desktop Intelligence) : Le Cache Server est responsable du traitement de toutes les requêtes de visualisation de rapport. Le Cache Server vérifie s'il lui est possible ou non de répondre à la requête en affichant une page de rapport mise en mémoire cache. S'il ne peut répondre à la requête en proposant une page de rapport mise en mémoire cache, le Cache Server transmet la requête au Page Server.

4. Event Server : Gère les événements basés sur des fichiers. Il avise le CMS du déclenchement de l'événement basé sur un fichier.

5. Job Server : Traite les actions planifiées sur des objets à la demande du CMS. Lorsque vous ajoutez un Job Server au système **BusinessObjects Enterprise**, vous pouvez configurer le Job Server pour : le traitement des objets rapport (Report Job Server), le traitement des objets programme (Program Job Server), l'envoi des objets ou instances vers des destinations spécifiées …

6. Input File Repository Server : Gère tous les objets de type rapport et programme ceux qui ont été publiés dans le système par des administrateurs ou des utilisateurs finaux.

7. Output File Repository Server : Gère toutes les instances de rapport générées par le Report Job Server ou le Web Intelligence Report Server et les instances de programmes générées par le Program Job Server.

8. Report Application Server (RAS) : Traite les rapports que les utilisateurs visualisent à l'aide du visualiseur DHTML avancé. Le RAS fournit également des fonctions de reporting ad hoc qui permettent aux utilisateurs de créer et de modifier les rapports sur le Web.

9. Connection Server

stopservers

Le script **stopservers** se trouve dans le répertoire « **bobje** » de votre installation. Ce script peut être utilisé comme modèle pour vos propres scripts : Il est fourni comme exemple pour montrer comment vous pouvez configurer votre propre script permettant d'arrêter les serveurs de **BusinessObjects Enterprise** en exécutant une série de commandes dans le CCM.

Exemple : **# ./stopservers**

<u>Liste des Procédures</u>

Arrêt/Redémarrage du CMS

Si vous n'arrivez pas à vous connecter au cms à l'aide des outils BO (Web Inetlligent – Desktop Inetlligent...).

Vérifiez que votre serveur cms est accessible (reseaux tcp/ip fonctionnel) en faisant un test de ping sur l'adresse ip du serveur.

Vérifiez que le serveur de déploiement, Tomcat dans notre cas, est démarré. Pour cela, saisissez l'adresse url suivante dans un navigateur :

http://nom_serveur:numero_port ou http://adresse_ip_serveur:numero_port (ex http://Fred:8080). Sinon localisez le fichier tomcatstartup.sh situé dans le répertoire « **bobje** » du répertoire d'installation de BusinessObjects. Accédez à $BO_HOME/bobje/ et exécutez tomcatstartup.sh comme suit # ./tomcatstartup.sh

Vérifiez que le cms est démarré sinon exécutez le script :

ccm.sh –nom_cms start (démmarrage du cms)

Ou

./stopservers (arrêt de tous les services)

Puis

#./startservers (demarrage de tous les service)

Si vous avez choisi une installation utilisateur, vous pouvez activer automatiquement les scripts de démarrage/arrêt des serveurs BusinessObjects (startservers – stopservers). Il suffit pour cela de taper la commande suivante (en root, bien sûr !) :

chkconfig --add nom_du_script # chkconfig --level 345 nom_du_script

Il est également possible quoique déconseillé de démarrer l'application en configurant le fichier /etc/rc.d/rc.local, qui est exécuté à chaque démarrage de la machine.

En résumé, pour qu'une application soit lancée au démarrage d'un serveur Linux et que l'on puisse l'utiliser, il faut que l'application soit installée que l'on ait le droit d'y accéder que son démarrage soit prévu dans le fichier /etc/inetd.conf (ou dans le répertoire /etc/xinetd.d), dans un script situé dans /etc/init.d (avec un lien symbolique dans le niveau de démarrage souhaité, ex : /etc/rc5.d pour le mode graphique) ou bien encore dans le fichier /etc/rc.d/rc.local si et seulement si le programme est lancé par inetd/xinetd, alors le numéro de port doit être présent dans /etc/services.

Sauvegarde

Nous recommandons de sauvegarder la base du référentiel BO avec une fréquence journalière.

DOCUMENTATION DU PROJET

Souvent le parent pauvre d'un projet, la documentation est pourtant nécessaire. Une documentation tenue à jour est chose rare de nos jours parce que cela coûte cher, que les évolutions sont souvent rapides, que l'on compte trop souvent sur sa mémoire, que cela n'est pas valorisant de le faire, etc. etc. Nous pourrions trouver encore mille excuses.

Je vous propose une liste, elle n'est pas un aboutissement, mais une base de travail que vous pourrez compléter selon votre organisation, client, vos taches ou autre.

Ces chapitres contiennent les informations nécessaires à la réalisation d'un besoin fonctionnel ou technique.

SPECIFICATIONS FONCTIONNELLES

Élaborées et alimentées à partir du dossier d'analyse et des réunions de travail, elles décrivent notamment :

- Le métier et l'organisation.
- L'ensemble des fonctionnalités attendues.
- Le dictionnaire de l'ensemble des données.
- Les règles de gestion utilisées.
- Une cartographie fonctionnelle.
- Le schéma conceptuel et physique des données.
- La description de l'ergonomie et de la navigation de l'interface.
- …

ARCHITECTURE APPLICATIVE

Doit permettre une vision de l'architecture applicative du projet. Il comporte notamment :

- Schéma général de la solution.
- Description des modules fonctionnels identifiés et de leurs relations ou interaction.
- Description technique des composants.
- Les normes et standards et de programmation utilisées.
- Règles de conception utilisées.
- L'environnement de développement.
- Une présentation des modules fonctionnels identifiés et de leurs relations.
- …

ARCHITECTURE TECHNIQUE

Doit permettre une vision de la solution proposée correspondant à l'environnement de production définitif. Il comporte notamment :

- Présentation générale de la solution.
- La description de l'architecture logique et physique (flux, matériel, réseau, logiciel…).
- Les outils et composants utilisés pour l'exploitation.

- Une étude de dimensionnement applicative.
- Les ressources techniques nécessaires (puissance, stockage, bande passante...).
- …

JUSTIFICATION

Ce document est généralement produit lorsqu'il y a nécessité de justifier des orientations fonctionnelles ou techniques différentes de celles proposées ou que la conception du périmètre retenu aboutit à proposer plusieurs scénarii de solutions. Ce document justifie chaque scénario proposé. Il comprend notamment :

- L'architecture fonctionnelle
- L'architecture technique.
- Les impacts identifiés ou non prévus à l'origine.
- Les prérequis identifiés ou non prévus.
- Une analyse des avantages et des inconvénients par rapport à l'existant.
- …

LE DOSSIER DE REALISATION

LE REFERENTIEL FONCTIONNEL

Ce référentiel à minima :

• Les données :

- Dictionnaire des données.
- Règles d'intégrités.
- MCD, MPD ou leur équivalent.

• La description fine des tables, fichiers, univers utilisés comprenant en particulier :

- les index mis en place.
- les procédures stockées, triggers et règles d'intégrités référentielles implémentées.
- les éventuelles codifications adoptées pour représenter certaines données.

• Les traitements.

- Règles de calcul.
- MCT, MOT ou leur équivalent.
- Scripts d'alimentation.
- Processus.

• Les flux de communication.
• Référentiels structurants.

- Maquettes.
- normes.

• Chartes :

- Graphiques et d’identification.
- De navigation et d’ergonomie.

LE REFERENTIEL TECHNIQUE

Cette documentation décrit en particulier pour chaque application :

• Les algorithmes particuliers utilisés.
• La répartition des modules dans les répertoires et fichiers contenant le code source.
• Les procédures de création des exécutables à partir du code source.
• Les procédures de création du « Master » de diffusion.
• Scripts de création de l’ensemble des composants applicatif ou base.
• Paramétrage et configuration particulière des composants de développement utilisés.
• Le code source documenté des modules.
• Les logiciels sous forme exécutable.

PLAN DES TESTS UNITAIRES

Pour chaque module et pour chaque assemblage de modules qui a fait l’objet d’un test de la part du titulaire, ce document les références :

• Les jeux d’essai.
• Fiche de test.
• La date de passage du test.
• Le résultat de l’exécution du test.

Pour toute nouvelle version développée, le titulaire doit fournir, le plan de test de l’application comprenant :

• Les dossiers de tests unitaires.
• L’outillage de test.
• Les modules de programmation utilisés pour les tests livrés sous forme de sources et d’exécutables (objectifs de reproductibilité).
• La documentation de réalisation (description des méthodes).
• Le compte-rendu de la campagne de tests.

LE DOSSIER DE RECETTE

Ce document expose l’ensemble du périmètre testé par le titulaire. Il est complété par la suite par l’équipe chargée de la recette. Ce document est complété par des scénarii de tests du plan de test. La structure de ce cahier peut se présenter de la manière suivante :

• Organisation.

- Périmètre fonctionnel.
- Périmètre technique.

- Modalités.
- Scénarii de test.

• Plan de test (pour chaque fonction).

- Localisation (interface).
- Description.
- Périmètre (référence à une fiche de test).
- Modalités.
- Données.
- Règles de gestion.
- Impact(s) identifié(s).
- Résultat attendu

LE DOSSIER TECHNIQUE

• La procédure et notice d'installation.
• La procédure et notice de désinstallation.
• Les procédures de construction de l'exécutable à partir du code source.

MANUEL D'INSTALLATION ET DE DESINSTALLATION

Ce document précise la procédure à pour notamment :

• Installer un outil.
• Installer un composant.
• Installer l'ensemble applicatif.
• Paramétrer et configurer les plates-formes à recevoir l'ensemble applicatif.
• Recommandation de désinstallation particulière.
• Des procédures liées à l'installation.
• Des procédures liées aux paramétrages et configurations.
• Recommandations.

MANUEL D'EXPLOITATION

Pour chaque chaîne d'exploitation du serveur sera décrit :

• Les fichiers en entrée.
• Les fichiers en sortie.
• Les ressources nécessaires (place disque, réseau...).
• Les paramètres d'exploitation.
• Le dialogue exploitant qui l'accompagne.
• La gestion des points de contrôle et de reprise.
• La possibilité de lancer ou non la chaîne durant l'exploitation du transactionnel.
• Le langage de commande de la chaîne.

Les utilitaires divers (réorganisation de base, sauvegarde/restauration de la base, ajout/suppression de terminaux, gestion des habilitations...) seront également décrits.

Pour chacun de ces utilitaires, on décrira le dialogue nécessaire à sa mise en œuvre.

GUIDES D’EXPLOITATION

Décrit l’ensemble des procédures nécessaires à l’administration et à la maintenance des serveurs et des applications et outils mis en exploitation.

Cette documentation décrit les tâches quotidiennes de maintenance et de surveillance de fonctionnement du système, d’administration et de gestion de l’authentification. Il est accessible uniquement aux administrateurs systèmes et équivalents. Pour chacune des procédures, on décrira le dialogue exploitant nécessaire qui l'accompagne.

Ce document traite notamment :

• De l’administration et des procédures d’optimisation (configurations, paramétrage, Tunning, …)
• Des procédures liées à la sécurité et à l’accès.
• Des procédures de sauvegarde et restauration.
• Des procédures d’arrêt et de redémarrage.
• Des scripts et procédures de création et d’alimentation.
• …

Pour chaque chaîne d’exploitation ou d’alimentation sera décrit :

• Les fichiers en entrée.
• Les fichiers en sortie.
• Les procédures de lancement et d’arrêt et leurs paramètres et le langage de commande qui les accompagne.
• La gestion des points de contrôle et de reprise.
• La gestion des erreurs.
• L’organisation recommandée dans le cadre de l’exploitation.

DOCUMENTS D’UTILISATION

Le manuel utilisateur décrit dans le détail les différentes fonctionnalités de l’application et les règles de gestion correspondantes y sont décrites. Cette documentation décrit l'utilisation des menus, les dessins et l'enchaînement des écrans, les contrôles et calculs effectués. Elle explique les messages d'erreur et actions à entreprendre en cas d'anomalie.

Elle est remise à l'administration sous forme électronique ou sous la forme d’un fichier de type « aide Windows ».

• Structure de l’application et enchaînement des écrans.
• L'utilisation de l’interface et des menus et les principes génériques de navigation et d’ergonomie.
• Le détail les différentes fonctionnalités de l’application et leurs utilisations (interface, traitement, résultat).
• Les règles de gestion correspondantes.

• Les contrôles et calculs effectués.
• Les erreurs possibles et l’explication des messages rencontrés et les actions à entreprendre dans ce cas.
• …

LE DOSSIER DE MAINTENANCE

RAPPORT DES ACTIONS CORRECTIVES

Ce support doit notamment faire apparaître :

• La liste des anomalies rencontrées.
• La liste des anomalies en cours de correction et leurs états.
• L'origine de chaque anomalie.
• Les solutions de contournements mis en place.
• La solution corrective envisagée.
• L'impact formalisé de la solution qu'il propose sur la version du composant.
• La liste des constats ayant fait l’objet de la mise en place d’une solution de contournement (notamment, les anomalies bloquantes en phase de recette).
• Le délai de résolution des constats clos et non clos.
• Un planning prévisionnel de correction.

RAPPORT DES ACTIONS EVOLUTIVES ET ADAPTATIVES

• Liste des actions.
• Périmètre.
• Impact(s) sur l’application

LE DOSSIER DE PILOTAGE

Ces documents sont des supports de travail interne au groupe projet client. Ils rassemblent les informations suffisantes au suivi du projet et sont principalement composés du :

COMPTE-RENDU DE REUNION

Ce document de suivi de projet doit faire apparaître notamment :

• Les décisions prises et leurs conditions de mise en œuvre (quoi, qui, quand).
• Un suivi des points d’action.
• Les questions restées sans réponse.

RAPPORT D’AVANCEMENT (OU SYNTHESE DE PROJET)

Ce document de suivi de projet doit faire apparaître notamment les estimations des coûts et délais de réalisation et une analyse des écarts comprenant :

• Analyse des risques et les causes des dysfonctionnements.
• L’estimation des coûts et du reste à faire.
• Les charges.

- La planification générale.
- les répercussions sur le calendrier général des travaux (retards ou avances calendaires).
- Bilan des points d’action
- les éléments de base permettant de déterminer les actions correctives.
- Les mesures préventives ou correctives, proposées.
- Le travail planifié pour la période suivante.

BILAN DE PROJET

Document ultime du projet, il rassemble les évènements marquants du projet et permet au client de tirer des conclusions sur l’ensemble des actions réalisées. Il fait apparaître principalement les difficultés rencontrées et les solutions :

- Bilan de conception.
- Bilan de réalisation.
- Bilan de mise en production.
- Bilan de déploiement.
- Bilan global du projet (principalement sur les écarts rencontrés).

Heureusement, il est rare que l'on vous demande de fournir toutes ces documentations au cours d'un seul et même projet, mais c'est possible, cela m'est déjà arrivé.

Gestion de projet **informatique**

On parle tous de projets dans la vie de tous les jours : nos projets de vacances, projets de carrière, projets d'avoir des enfants... Le terme projet est donc un terme du vocabulaire courant, auquel on associe une signification relativement claire et précise : c'est un ensemble d'actions que nous souhaitons entreprendre, pour atteindre un but (devenir parent, embrasser une nouvelle carrière...). En ce sens, le projet est bien « le brouillon de l'avenir » : une ébauche, mais pas encore une réalisation.

La gestion de projet (ou conduite de projet) est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Lorsque la gestion de projet porte sur un ensemble de projets concourant à un même objectif, on parle de gestion de programme.

Au Sommaire

Le cycle de vie d'un projet • Modèles de développement • Modèle en cascade • Modèle en V • Modèle en W • Modèle incrémental • Modèle évolutif • Modèle de cycle RAD • Modèle en spirale • UP • RUP • Le cycle de l'Extreme Programming • Répondre à un appel d'offre décisionnel • Facteurs clés du succès • Facteurs clés en mode liste • Recueil de besoins • La planification • Solutions décisionnelles • Technique et maîtrise d'œuvre • Projet décisionnel • Spécifications • Facteurs de succès • Logiciel décisionnel • Business Objects • Documentation du projet • Architecture applicative • Architecture technique • Le dossier de réalisation • Le référentiel fonctionnel • Le référentiel technique • Plan des tests unitaires • Le dossier de recette • Le Dossier technique • Manuel d'exploitation • Guides d'exploitation • Documents d'utilisation • Le dossier de maintenance • Rapport des actions correctives • Rapport des actions évolutives et adaptatives • Le dossier de pilotage • Compte-rendu de réunion • Rapport d'avancement (ou synthèse de projet) • Bilan de projet

Stéphane Grare

Concepteur et développeur C#, Stéphane Grare, passionné par la programmation informatique, est l'auteur de plusieurs tutoriels et livres blancs sur différents langages de programmation informatiques. Il est aussi à l'initiative du projet Simply, une gamme d'applications dédiées à la bureautique.

À qui s'adresse cet ouvrage ?

- Aux étudiants de la formation Concepteur Développeur
- Aux étudiants de la formation Développeur Logiciels

Sur le site www.afpa.fr



- La formation professionnelle
- Valider ses acquis
- Se remettre à niveau

Sur le site system-sg.perso.sfr.fr